UiO : Department of Informatics
University of Oslo

# Measuring and Comparing the Stability of Internet Paths over IPv4 & IPv6

Using NorNet Core Infrastructure

Forough Golkar
foroughg@ifi.uio.no

Master's Thesis Spring 2014

# Measuring and Comparing the Stability of Internet Paths over IPv4 & IPv6

Forough Golkar
foroughg@ifi.uio.no

20th May 2014

*Dedicated to my lovely mother, Fariba*

# Abstract

Enormous Internet growth is an obvious trend nowadays while Internet protocol version 4 address space exhaustion led into the invention of Internet protocol version 6 and subsequently raised the transitioning challenges from one protocol to the other one. As the Internet protocol version 6 have not been around for a long time, thus there is a limitation knowledge in topology, routing change and performance of IPv6. Several studies are going on in order to getting the information about IPv6 which makes a contribution with improvement and deployment of IPv6 and accelerates this transitioning which one of the aspects is the Internet path stability.

With this aid this thesis proposed a comparative study of Internet path stability over IPv4 and IPv6. First a measurement of Internet paths over both IP versions has been performed, then the Internet path changes have been extracted in each single set of source-destination pairs. Finally some analysis on the data has been performed in order to observe the stability of Internet paths and provided a comparative study in the set of IPv4 and IPv6.

# Acknowledgement

I would like to express my appreciation to the following persons for their kind supports during this thesis:

- My sincere gratitude to Amund Kvalbein, for accepting supervising me. He is not only a great professor but also a wonderful person. Thanks for all his supports and helps, motivation, encouragement and his guidelines to overcome the challenges faced throughout this thesis.
- My appreciation to Thomas Dreibholz my other supervisor, for all his efforts and supports. For his kindness helps in both technical and scientific way during this thesis.
- My deep gratitude to my internal supervisor, Paal Engelstad, as a knowledgeable professor. For his supports, thoughtful advices and encouragements during this thesis.
- Special thanks to Hårek Haugerud as a great and wonderful person and professor, all his supports, encouragement and kindness during these two years of master study. Moreover, for trusting me to be as teaching assistant in Intrusion detection and rewall course.
- Thanks to Kyrre Begnum as a knowledgeable professor, and great person. Words can not describe the amount of technical and scientific stuff that I learned from him.
- Thanks to Ismail Hassan for all his help and so many technical stuff that taught us in this master degree.
- Thanks to Anis Yazidi for all his advices and supporting me as a teaching assistant in Scripting for system administration course.
- My special and immeasurable thanks to Mehraj Pakmehr who is not only a classmate but also a wonderful friend with all his support and help during this two years of master degree. His kindness is not expressible by words.
- I would like to say thanks to the University of Oslo and Oslo University College for offering this Master degree program.
- Thanks to all my friends and fellow classmates for their company during this two years.

- Last but not least, I would like to thank my beloved family including mother, father and brother. I could not be who I am and at the place I am right now without their total support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The tremendous success and growth of Internet from a few decades ago proved the scalability and robustness of Internet protocol (IP). The extensive usage of Internet by Individual humans, from personal computers, mobile devices and tablets; as well as enterprises with verity size of networks provide a globally interacting community. The stabilize growth of Internet made a communication and interconnection pattern which today no one can imagine one day without Internet connectivity and the catastrophic impacts on military, health care, e-commerce services.

The expansion Internet growth led to Internet protocol Version4 (IPv4) [66] address space exhaustion on February 2011[37]. The exhaustion of IPv4 has been known to happen for a long time, hence the invention of Internet Protocol Version6 (IPv6) [15] was the only solution in order to support the ongoing Internet growth. Due to the non-compatibility of IPv6 with IPv4, both networks cannot communicate with each other naturally therefore both protocols are needed to work in parallel for a long period until the complete IPv6 implementation. Despite the demand for availability of both IPv4 and IPv6 resulted in slow speed of transitioning. As the reason of IPv4 addresses are running out at this moment which causing an immediate need for IPv6 addresses, thus this process became an undeniable choice since a few years ago and the extreme pressure of IPv4 shortage forced researchers and Internet engineers to pay a significant attention in order to deploy IPv6 in nature[23, 82].

Today the statistics shows that the IPv6 ASes[1] following an exponential growth ($y = ae^{bx}$) over the past 10 years while IPv4 ASes following the linear growth ($y = ax + b$) as can be seen from Figure 1.1. Although the growth trend of IPv6 today resembles IPv4 growth trend at the early years of Internet emerge. The Internet Engineering Task Force (IETF) publish IPv6 in 1998 and now after 16 years still the number of adaptation of IPv6 is still much less than IPv4. Today the number of available IPv6 ASes is *8047* compare to the *46410* number of IPv4 ASes [16, 17, 38].

[1] AS(Autonomous System) is a single or group of network that controlled by a common network policy and represent a unit of routing policy.

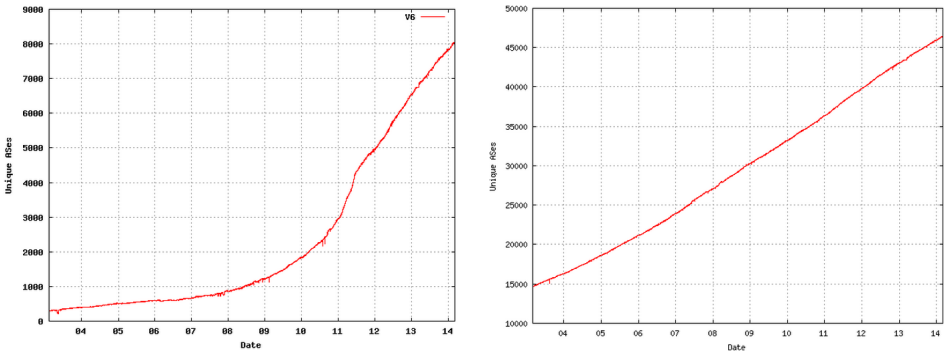



Figure 1.1: IPv6 Ases vs. IPv4 ASes [38].

It has been shown that the IPv6 maturing over the couple of years ago, and the progressing trend of IPv6 adoption is an inspiring sign. From the topological sight of view the deployment of IPv6 is mostly in the core of the network such as vendors or network equipment providers which selling network equipments, and Internet service providers(ISPs) or such a businesses that want to expand their networks. Therefore they need more IP addresses and since the IPv4 has been ran out, they have to deploy IPv6 in order to serve additional customers, otherwise they will lose their customers.

In the counterpart, there is lack of IPv6 adoption in the edge of network which widely consists of content providers, enterprise customers and residential users. Now just *3.8%* of websites are using IPv6 consist of some famous service providers and content providers like Google, Facebook, Youtube, Bing, Wikipedia, Yahoo and etc. [75]. As can be seen from graph 1.2 Google reported IPv6 adoption statistic that just *2.73%* of users connecting Google over IPv6 [31].

The Network Address Translation (NAT) technology is made an immediate IPv6 adoption less appealing in some networks. But NAT have some drawback consequence in a way that reduce network performance and increase network complexity. NAT also breaks end-to-end connection in nature and disturbs such a real time application like streaming. In addition some networks deploy tunneling or any kind of IPv6 translation in order to provide accessibility to IPv6 networks which these technologies can result in security exposure or unauthorized data access [24, 45].

One of the reasons behind slow IPv6 adoption trend can be the lack of incentive. Lack of encouragements for enterprise customers or end-users to deploy IPv6 is because of the IPv6 adaptation includes some expenses such as new a equipment, training staff, management and troubleshooting overhead; out of getting any concrete benefit back[17]. On top of that the lack proliferation of application for IPv6 which can encourage or force end


Figure 1.2: Google IPv6 adoption statistics report. [31].

users or enterprises to deploy IPv6 can also lessen the end users willingness to deploy IPv6. In this way users do not feel that they will miss anything without using IPv6.

Lack of knowledge about IPv6 performance and end-to-end support of IPv6 in nature is another reason of low speed adaptation.

Many items can have influence on performance such as network equipments and network policies and technologies which are used in IPv6 implementation. The ASes and ISPs network policies are not much known because they reluctant to reveal their network infrastructure and policies or do not keep the registry up-to-date[30], hence in the positive sight of view there can be a probability of existence of IPv6 network with higher performance in some Networks which IPv6 have had more prioritize which resulted in using new and cutting-edge network equipment and technologies in order to provide better performance. On the other hand and at the negative sight of view there is likelihood which some networks deploy IPv6 by using some non-effective tunneling or any kind of translation technology with low network capacity which yields in worth performance.

Another significant entity that has a great influence on network performance and network availability is path stability. **Internet path** is defined as the series of intermediate Autonomous Systems(ASes) between source and destination routers which create a routing path. The instability of path has a direct impact on the end-to-end Internet performance and availability. **Routing instability** is informally defined by the regular or fast changes of network reachability and topology information which led in Internet path fluctuating. This instability might be as the result of router configuration error, temporary physical and data-link problem which affect routing pro-

tocol problems and results in routing instability [42, 44, 46] like route loops, route fluctuations and synchronization [68].

Each network consist of Data plane and Control plane parts, which data plane is responsible for forwarding packet to the destination and control plane is specifying the route. When a network changes, the old nonexistence route will be removed, new route will be determined and advertised. These changes cause instability in network control plane. Instability in control plane due to the network changes derive instability in data plane that cause poor end-to-end performance and reduce the Internet infrastructure efficiency [72]. Packet loss and network latency increment are affected from high level of path instability, in addition the excessive low level of stability result in loss of Internet connectivity [43, 79].

One of the widely used transport layer protocol which is directly affected from the path instability is Transmission Control Protocol (TCP) [65]. TCP use retransmitting technology in the situation that the sender does not receive any acknowledgment within a particular delay time (retransmission timeout-RTO). In this condition TCP assumes any packet loss is because of network congestion, therefore TCP triggers its congestion control mechanism to subtract the network congestion. Depends on the congestion control mechanism, the congestion window is reduced and sender backs off in order to keep the throughput high and minimizing the packet loss which led in the low performance and efficiency of the TCP [11, 68, 81].

Researchers have been measuring IPv4 characteristics for a long time and the are a lot informative knowledge in hand about IPv4 such as ASes topologies, routes changes, performance, stability and so on . Since IPv6 was not available until a few years ago, there are not a lot information about it. At this moment several researches threads going on around the world to know IPv6 and comparing with IPv4, hence one of the aspect is the stability of Internet path.

## 1.1 Problem statement

This study focus on measuring the Internet path stability by using NorNet Core infrastructure, evaluate the stability of IPv6 in the Internet and comparing with the stability of IPv4. This thesis attempts to implement an appropriate measurement with the purpose of applying accurate comparison. This data measurement is intended to help NorNet experimenters in a way that providing some information about the underlying path while the person how conducting experiment in NorNet Core testbed can only see the tunnels between endpoints. This study also strives to increase the value of experiments using NorNet testbed. This report tries to address following research question:

- *Characterise the stability of end-to-end Internet paths over IPv6, and*

*compare it to of IPv4 paths*

  - *Active, long-term measurement of Internet route changes in the set of IPv4 and IPv6 paths.*

- *Analysing the nature of Internet path changes.*
  - *Distinguishing the real Internet path changes from load balancing.*
  - *Measuring the temporal characteristics of path changes. Do they happen in burst or continusly?*

## 1.2 Thesis Contributions

As an attempt to improve the Internet path stability knowledge with the purpose of finding the advantages and flaws of IPv6 deployment, the focus of this thesis is on the measuring the Internet path stability over IPv4 and IPv6.
This thesis makes the following contributions:

We develop a tool within the Nornet Core infrastructure for monitoring the paths between NNC endpoints. The tool gives experimenters in NNC information about the quality and dynamics of the underlying Internet paths, which is useful in many scenarios.

We make the collected information available in a central database, accessible to all NNC experiments.

We perform a study of path stability over IPv4 and IPv6 between endpoints in NNC. Based on 7 weeks of collected data, we make four main observations:

- IPv6 paths are longer than comparable IPv4
- There are more load balancing in IPv6 paths than in comparable IPv4 paths.
- IPv6 paths are less stable (experience more changes) than comparable IPv4 paths
- IPv4 paths to remote locations experience significantly more path changes than paths within Norway.

## 1.3 Thesis Outline

The structure of this thesis followed by:
**Chapter 1** (Introduction) which provides an overview of the extensive Internet growth and it's importance in today's life. Further this chapter illustrates some reasons behind the slow trend of IPv6 adoption and the importance of path stability and it's effect on IPv6 performance. In addition the questions which are trying to answer in this thesis and problem statement is defined.

**Chapter 2** (Background) firstly describes the story behind IPv4 exhaustion and the IPv6 history. Secondly this chapter provides an overview of NorNet Core testbed infrastructure and describes how it works. Furthermore this chapter illustrates the brief description of some network measurement tools. Finally some previous attempts represents as related works in this chapter.

**Chapter 3** (Approach) propose the whole design of this research from the measuring Internet paths over IPv4 and IPv6 and data collection, centralizing collected data, approaches of data processing and analysis design.

**Chapter 4** (Implementation) describes the whole implementations of measurement service and data processing procedures based on the design in the approach chapter.

**Chapter 5** (Results & Analysis) illustrates the exact observations out of the data processing and the explanations of each analysis steps.

**Chapter 6** (Discussion & Future Works) demonstrates the overview of this thesis work, discuss the pitfalls and possible improvements and express some future works.

**Chapter 7** (Conclusion) shows the actual outcome and observations out of the performed analysis in the stability of Internet paths perspective.

# Chapter 2

# Background

## 2.1 Transition To IPv6

The transition from Internet Protocol Version 4 (IPv4) to Internet Protocol Version 6 (IPv6) became a hot debatable topic since a few years ago. It is quite undeniable ongoing process nowadays while the rareness of IPv4 address capacity carried an extensive pressure to Internet Operators, vendors, Organizations, Operating Systems and network equipments in order to deploy and support IPv6.[17]

### 2.1.1 Exhaustion of IPv4 Addresses

The first recognition of IP address shortage in technical circles in August 1990[36] raised an argue between researchers and Internet operation engineers in order to find a solution to overcome this problem. The usage of IPv4 Internet become far larger than one could imagine at start stage of designing IPv4 protocol in addition the needs for Internet usage is still growing.

The IP addresses are allocated by Internet Assigned Number Authority (IANA) from unused IP address pool. IANA delegate IP addresses to the Regional Internet Registries (RIR), while Internet Service Provider (ISP) get IP addresses from Local Internet Registry (LIR) [35]. It became an obstacle in front of future technology progression in the position that the Availability of unused IP addresses becomes zero. Therefore the shortage problem of IPv4 address space result in the design of newer Internet protocol known as Internet Protocol Version 6 (IPv6) in 1998[33].

IPv6 has been design to overcome the biggest IPv4 problem which is the scarcity of Internet addresses space and some other architectural IPv4 limitation such as routing scalability and end-to-end property.

Over the last couple of years the problem has gone from a theoretical one to practical one since IANA reported the exhaustion of IPv4 address and allocated the last five/8 address block to the RIRs on February 2011[37]. This process is followed by Asia-Pacific region (APNIC RIR) as

well on April 2011[2] and RIPE was the next that started to allocating last/8 on September 2012[69]. The other RIRs also going to be exhausted in few years[37].

Now researcher communities and Internet operation engineers put a significant consideration to improve the trend of transition from IPv4 to IPv6.

### 2.1.2 IPv4 vs. IPv6 Protocol Specification

IPv6 and IPv4 have some differences in some categories such as:

- Addressing:

  The first reason behind IPv6 creation was the IPv4 address limitation which causes the IPv6 address space to be large enough that can support the fast speed of Internet growth and large enough for the anticipated future need. IPv6 support 128-bit address length which be able to supply approximately $3.4 * 10^{38}$ addresses while IPV4 with 32-bit address length just can provide $3.4 * 10^{9}$ addresses which is not sufficient in order to support current world population[33]. The lesson from IPv4 address space limitation leads into the vast IPv6 address space that can cover the anticipated future Internet need. The intended design of IPv6 is to provide a usable lifetime of over 50 years compare to 15 years of IPv4 lifetime in which can encompass the hundreds of millions of new devices connecting to Internet every year[36].

  The 128-bit IPv6 address consists of 2 parts:

  - 64-bit network prefix: It provide great flexibility for network management while allocating /64 subnet size simplified address allocation compare to IPv4 with different subnet size. Furthermore IPv6 addressing remove the needs for NAT also ease the network renumbering with use of Router Advertisement (RA) [57] for changing prefix and Stateless Address Autoconfiguration (SLAAC) [74] for self-configuring interface identifiers.[33, 82]
  - 64-bit interface identifier: The Interface identifier is the same with subnet prefix which is fixed in IPv6 unlike to IPv4 which have different subneting, and is used for interface link identification.

- Address type:

  The IPv4 supported unicast, multicast and broadcast while the broadcast fortunately has been removed in IPv6 which had a lot of problem with some networks and the multicast is used instead. Anycast addresses is a new from RFC 1546[9], It is used in IPv4 and IPv6.

- IP Header:

  IPv6 have simpler header compare to IPv4. IP Header Length(IHL), Type Of Service(TOS), Checksum and also fragmentation-related field has been removed or moved to optional extension header. The IPv4 header and IPv6 header has been shown in Figures2.1 2.2 respectively.

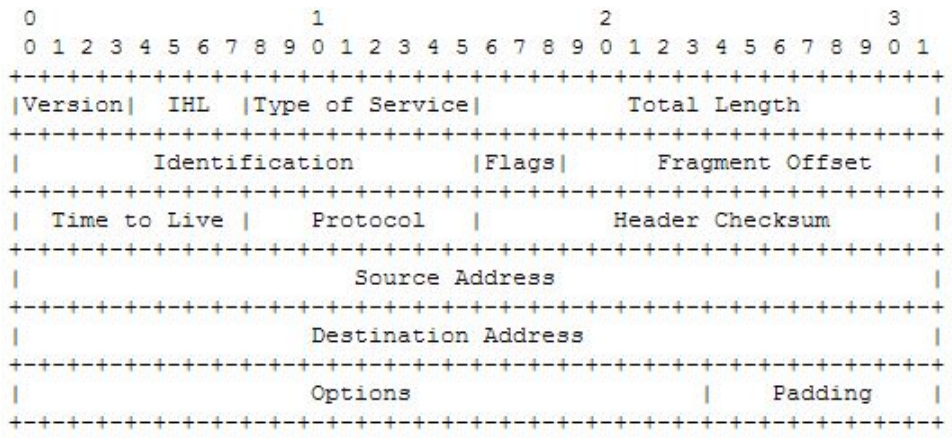



Figure 2.1: IPv4 header format [66].

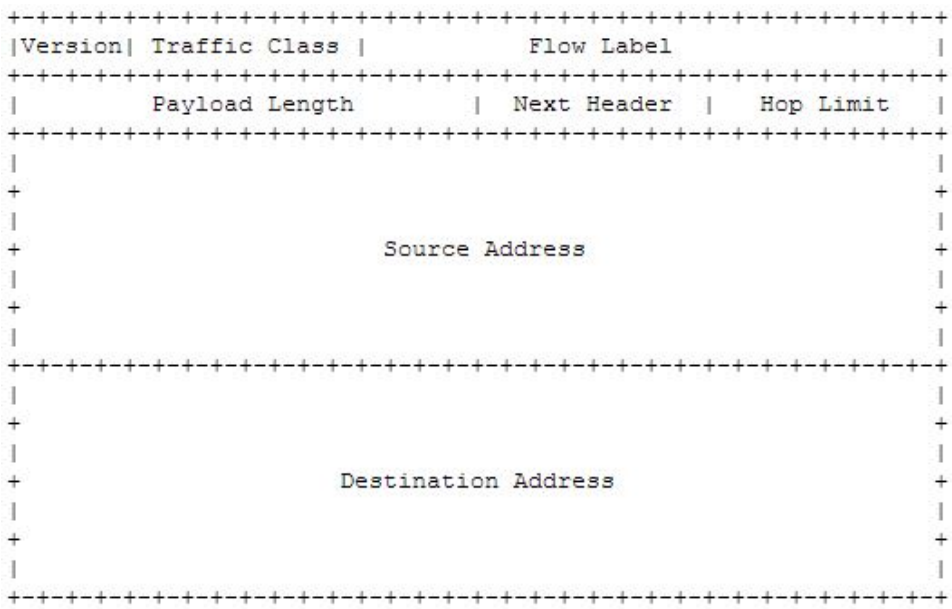



Figure 2.2: IPv6 header format [15].

## 2.2 NorNet Multi-Homed Research Testbed

Nowadays the large global testbed platforms became important as the fundamental progress point of future Internet. These testbeds designed to propose the realistic, flexible and capable infrastructure for researchers to execute tests and measurements. NorNet is one of the open, large scale and distributed testbeds with multi-homed connections, while the multi-homing feature is the connection of each NorNet site to multiple Internet Service Provider (ISP) or in other word the connection to more than one ISP in each site, in order to provide realistic, constant and robust Internet connectivity. The goal of NorNet project is to supporting network research experiments while obviously these kind of tests must be performed on a capable platform to capture the network traffics, find failure points and bottlenecks in real network. On the other hand such these testing cannot performed in production networks because of the influence of them on the stability of network. Hence by the use of NorNet, the researchers can get the benefit of multi-homing testbed with high availability in order to conducting experiments and measurements. [25] Each Internet Service Provider (ISP) of each site act simultaneously and independently in a way that if connectivity to any of ISPs fails, the connection to the other one still available. NorNet is built by Simula Research Laboratory.

NorNet has two components: First one is NorNet Core which is the wired part of testbed and wiry connected to Internet Service Providers, the second one is NorNet Edge which is the wireless part and connected to mobile broadband providers. The functionality of both parts will propose one of a kind platform for network researches [32].

### 2.2.1 NorNet Core

NorNet Core is the wired part of NorNet testbed and has been built on MYPLC software which is developed by PLANETLAB consortium. The PLANETLAB is in a debatable way large global testbed platform with 580 sites distributed all over the world for accessing globally network service [28, 60]. NorNet get the benefit of a large and well-maintained testbed, in addition it gets the advantages of multi-homing in order to increase the functionality of a global research testbed by providing multi-connectivity. At this time NorNet core made up of 14 program able sites which 11 of them are distributed in all part of Norway. 1 site is located in Essen-Germany, 1 site is placed in Karlstad-Sweden and another one is in Hanian-China.[27] Sites are mostly located on educational institute or research centers. The different locations of NorNet Core sites proposed in Figure 2.3. Each Site have at least 2 Internet Connection to different Internet Service Provider. The Sites with their geographical locations and their ISPs connection is presented in Table 2.1.

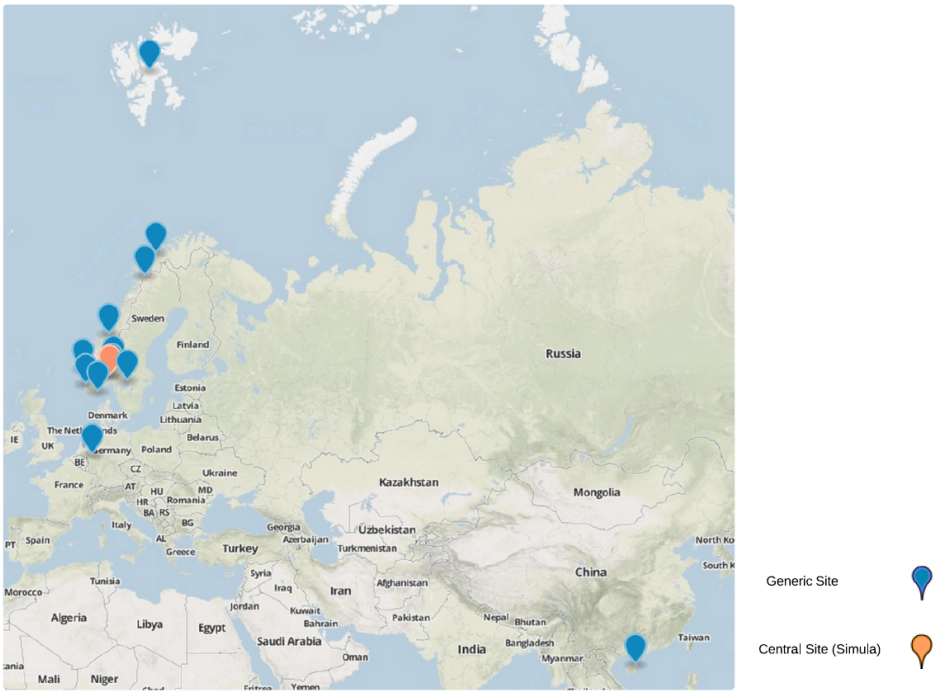



Figure 2.3: Current NorNet site map.

### 2.2.2 NorNet Core Architecture

NorNet Core set up tunnels in order to establish multiple connections between sites at the Network Layer. NorNet Core set up tunnel between NorNet Core site using static IP tunnel which is released by Generic Routing Encapsulation (GRE) protocol[29] over IPv4 and IPv6-over-IPv6 tunnels.[25]. NorNet core did not use any kind of Network Layer third-party VPN in order to prevent any side-effect of hidden routing, unnecessary comparison and encryption of VPN software since they cause extra load to the system which experiments running at.
Each sites of NorNet core made up of a set of nodes:

1. `Research Nodes`(PL, VINI,..) where experimenters run their experiments.
2. `Control Nodes`(ctrl) provide local access to the site such as local monitoring.
3. `Tunnelbox`(Tbox) is a router which manage all tunnels of site which make a connection to the other site. Tunnelbox represent the different combination of outgoing and incoming ISPs

At Simula site which is the central site of NorNet Core in addition to the research node, control node and tunnelbox, there is also `management`(myPLC) and `monitoring`(MYVINI) infrastructure as can be seen in Figure 2.4.

Each site is connected to at least to 2 ISPs. If we can consider such an example of connection between one pair of NorNet Core sites in a condition

| Site | Location | ISPs |
|---|---|---|
| Simula Research Labratory | Fornebu, Akershus | *1:* UNINETT<br>*2:* Kvantel (Hafslund)<br>*3: Telenor*[2] |
| University of Oslo | Blindern, Oslo | *1:* UNINETT<br>*2:* PowerTech<br>*3: Broadnet*[2] |
| Høgskolen of Gjøvik | Gjøvik, Oppland | *1:* UNINETT<br>*2:* PowerTech |
| University of Tromsø | Tromsø, Troms | *1: UNINETT*[1]<br>*2:* PowerTech<br>*3: Telenor*[2] |
| University of Stavanger | Stavanger, Rogaland | *1:* UNINETT<br>*2:* Powertech |
| University of Bergen | Bergen, Hordaland | *1:* UNINETT<br>*2:* BKK |
| University of Agder | Kristiansand, Vest-Agder | *1:* UNINETT<br>*2:* PowerTech |
| University of Svalbard | Longyearbyen, Svalbard | *1: UNINETT*[1]<br>*2: Telenor*[2,4] |
| NTNU Trondheim | Trondheim, Sør-Trøndelag | *1:* UNINETT<br>*2:* PowerTech |
| Høgskolen of Narvik | Narvik, Norland | *1:* UNINETT<br>*2:* PowerTech<br>*3: Broadnet*[2] |
| University of Duisburg-Essen | Essen-Germany | *1:* DFN<br>*2: Versatel*[2,3] |
| Hainan University | Haikou, Hainan-China | *1: CERNET*[1]<br>*2: China Unicom*[1] |
| Karlstads University | Karlstad,Vørmland-Sweden | *1:* SUNET<br>*2:* _4 |
| Høgskolen i Oslo og Akershus | St.Hanshaugen, Oslo | *1:* UNINETT |

1)IPv6 available from ISP, but not deployed to NorNet Core site
2) IPv6 not available from ISP
3) Consumer-grade ADSL connection
4) Negotiations in progress

Table 2.1: NorNet sites deployment status.

that site: $S_a$ is connected to 2 providers $P_{a1}$ and $P_{a2}$ and site: $S_b$ is connected to 3 providers of $P_{b1}$, $P_{b2}$ and $P_{b3}$, briefly:

$$S_a \rightarrow P_{a1}, P_{a2}$$

$$S_b \rightarrow P_{b1}, P_{b2}, P_{b3}$$

The NorNet Core make $|P_{sa}| \times |P_{sb}| = 2 \times 3$ tunnel boxes as can be seen from Figure 2.5. In addition tunnelbox setup tunnel for each IPv4 and IPv6 separately hence it create a fully connected tunnel mesh.

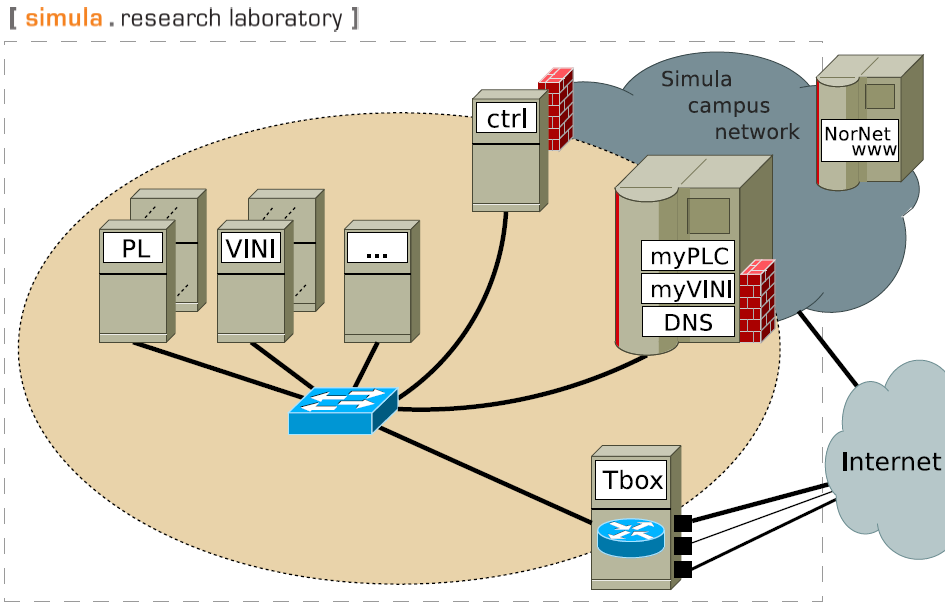



Figure 2.4: Simula Central Site Architecture [32].

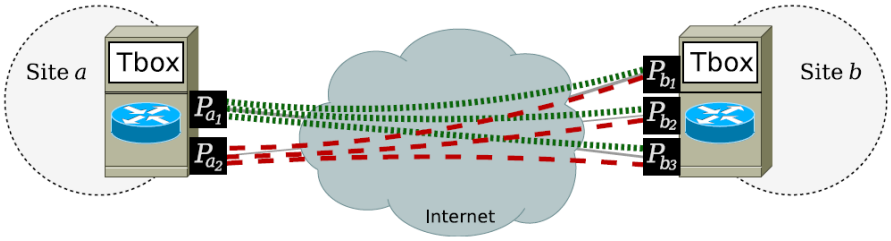



Figure 2.5: NorNet Core Tunnelbox[32].

### 2.2.3 NorNet Core Site Address Scheme

In the reason of IPv4 scarcity, furthermore because of the impossibility of allocating a single public IP address space per-Provider of each site to every other connected site; NorNet Core used private IP address space in each site which routing over public IP address through tunneling[32]. NorNet Core use a simple address scheme. For IPv4 :

10.<Provider Index>.<Site Index>.<Node Index>/24

and for IPv6:

2001:700:4100:<PP><SS>:<XXXX>:<NN>/64

while PP is two hexadecimal digit of Provider Index, SS is two hexadecimal digit of Site Index, XXXX is four hexadecimal digit which can be used as node-internal addressing and NN is two hexadecimal of Node Index [26, 32].

### 2.2.4 NorNet Core Tunnelbox

Tunnelboxes in NorNet Core manage the routing among the other tunnelboxes of NorNet Core sites. Since the NorNet Core tunnelboxes are Linux-based systems; NorNet project get the use of policy-based routing of Linux networking stack such as source address, Type Of Service(TOS) for IPv4[67] and Traffic Class Field for IPv6[15], and set up suitable routing table for IPv4 and IPv6, to make it possible to select the appropriate routing table based on packet source address instead of classic Internet routing which is based on the packet destination address.

### 2.2.5 NorNet Core Management

Since NorNet has been built based on PLANETLAB therefore NorNet use the same functionality as PLANETLAB software which provides database named *Planet lab Control* (*PLC*) with web-based configuration XMLRPC interface [58] called *PLCAPI* in order to managing users, nodes, and sites configuration. Each NorNet Core tunnelbox uses PLCAPI to fetch the configuration information from the Central *PLC* in order to setting up the interfaces[25, 32].

## 2.3 Internet Path Measurement Tools

In this section a review in the widely used Protocols and utilities for discovering and measuring Internet path is provided. Firstly a description about *Internet Control Protocol (ICMP)* and fallowing by *Ping* and *Traceroute* which are the two ICMP-based path measurement tools. The functionality and limitation of these utilities also provided as well.

### 2.3.1 Internet Control Message Protocol (ICMP)

The lack of reliability in Internet Protocol which is the network layer primary protocol may led into the packet duplication, data corruption and out of order delivery. For this reason some controlling mechanism is needed in order to avoiding such these fault events. Internet Control Message Protocol (ICMP) [14, 63] have been designed to diagnose and control the IP operation in the fashion of exchanging control or error message in the network such as routing control or packet processing error messages. ICMP is the dominant used Internet layer protocol and one of the part of Internet protocol suite. ICMP use the Internet Protocol (IP) to sending the *information Message* such as diagnostic and testing information or *error message* such as generated error in the packet delivery. These message notifications are known as *ICMP message*. Many widely used network utilities such as *ping* and *traceroute* are based on the ICMP protocol [6, 39].

ICMP also used to finding the path *Maximum Transmission Unit (MTU)* [55] which is preventing IP fragmentation. For example if a router cannot transfer a packet which is big in size and the *don't fragment* bit has been

set, sends a back an ICMP message of *"Too Big"* (Type 3, Code 4 with IPv4) message to the sender, therefore the sender reduce the packet size to the lowest path MTU. In the router perspective choosing the suitable path MTU will result in the better bandwidth [47, 49].

#### 2.3.1.1 How ICMP works

ICMP using IP header to sending *ICMP message* while each ICMP packet is 8 bytes of header and a variable size of payload or data section. ICMP is classified in *types* and *codes* which the types specified the ICMP function and *codes* of each specific *type* shows that *type* specification. The *ICMP type* field value in ICMP header which is illustrated in the Figure 2.6 define the type of ICMP message which varies from *0: Echo reply , 3: Destination unreachable, 11: Time Exceeded and etc*. Many of the ICMP types have specification *code* field, for instance ICMP type 3 which is the *Destination Unreachable* message have different specification *codes* such as *0: net unreachable, 1: host unreachable, 2: protocol unreachable, 3: port unreachable, 4: Fragmentation Needed and Don't Fragment was Set, and etc*. The complete list of ICMP type and codes are available in [63].

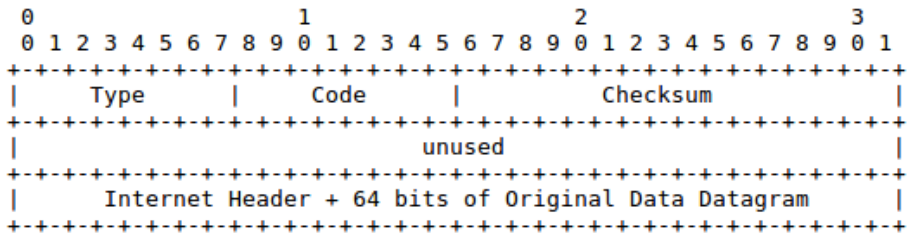



Figure 2.6: Internet Control Protocol (ICMP) Header for IPv4. [63].

ICMP is not used for the carrying the data, while it is a management protocol and is used in order check the status of the remote destination in the Internet by checking the availability of destination host and errors in the underlying network communication [76] .

#### 2.3.1.2 *ICMP* Issues

Hence ICMP have no authentication mechanism may result in to the attacks using ICMP. The *Denial of Service (Dos)* attack can be performed simply which results in communication break by copying the ICMP *error message* and sending to the both hosts in the communication. In addition some other attacks such as ping flood, ping of death and ICMP tunneling can be perform by or by using ICMP *information message*. The attacker can launch the DoS attack by forging *Time Exceeded* and *Destination Unreachable* message to the packet and stop sender connecting to the destination or stop destination responding to the sender. Attacker can also perform *Man-in-the-Middle (MiTM)* attack by *Router Redirect* message and sniffing the packets travelling between two hosts.

Some mechanism and utilities are available in order to detecting ICMP attack such as *Snort* which is a *Signature-based Detection System* or the *Router Configuration* mechanism which result in the router allows limited ICMP message types and drops the other ICMP message which is not satisfy the router configuration [6].

The Extensive ICMP attacks led into the restricted network firewall and router configuration and allowing ICMP traffic very cautiously. In some routers or networks firewall ICMP traffic control is so severe by denying all or significant ICMP traffic which brings some limitation for utility such as *ping* and *traceroute* that based on ICMP [6, 80].

### 2.3.2 Ping

*Ping* is the widely used networking tool in order to test reachability and availability of a remote host over Internet protocol. *Ping* is used by network and system administrators, researchers and users to find the network problems. It is an Internet Control Message Protocol (ICMP) based tool which is easy to use and can work universally in order to measure the end-to-end behaviour between two hosts.[61, 62].

Ping is written by `Mark Muuss` in 1983 as a troubleshooting tool which has been inspired by `Dr. Dave Mills` while he measured path latency using timed ICMP Echo packets [56]. It is available in all LINUX/UNIX based system and also ping6 is available for probing Internet Protocol version 6 (IPV6).

#### 2.3.2.1 How *Ping* works

*Ping* can be used in order to testing:

- Availability of remote host and checking whether it is active or inactive.
- Packet loss rate in the communication of the source and destination hosts.
- Latency or delay time between two hosts in a communication which is known as *Round Trip Time (RTT)*.

*Ping* is mostly used to testing whether a host is online or not. *Ping* sends an Internet Control Message Protocol (ICMP) probe as ICMP *echo request* to the destination host and waiting for the response or *echo reply* from the destination host. The source can get the response packet if the ICMP *echo request* probe could reached the destination and if the destination could be able to *reply* back within the predefined timeout which the default value of timeout in Linux is 2 RTTs.

#### 2.3.2.2 The basis of path latency measurement by *ping*

The time between sending a packet from source host until the packet is accepted in target host is known as one-way *latency*. If we consider the

Transmit time of packet as $T_{transmission}$ and the receiving time as $T_{receiving}$ the one-way delay is calculated by:

$$T_{transmission} - T_{receiving}$$

The traveling time of a packet between source to the destination and back is known as two-way delay or *Round Trip Time (RTT)*. [54]. As the figure 2.7 illustrates the source node timestamps when sending the *echo request* as $t_1$ and the $t_2$ is timestamped when the packet have been reached the target node. This is the same for *echo reply*, $t_3$ is the timestamps when the response packet leaves the target node and $t_4$ upon reaching the source node. The time between $t_2$ and $t_3$ is the time that the packet spent in the target node. $(t_4 - (t_1)$ is the time that the *request* packet leaves the source node until receiving the reply packet, with including the time that the packet spent in target node $(t_3 - (t_2)$. Therefore the RTT time is calculated by [40, 54, 83] :

$$RTT = (t_4 - (t_1) - (t_3 - (t_2)$$

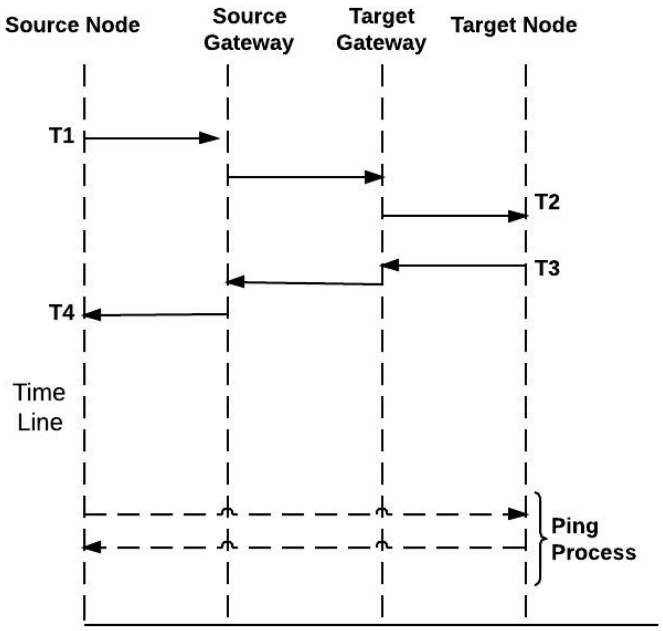



Figure 2.7: Round Trip Time Process i *ping*

#### 2.3.2.3 The Packet Loss Measurements by *Ping*

*Ping* observes the number of *echo request* probes that have been sent without receiving *echo reply*. The number of received packet compare to the number sent packet is known as the *packet loss rate* that is computing by the following formula:

$$packet - loss - rate = 1 - \left(\frac{packets - received}{packets - sent}\right)$$

In *ping* output the packet loss rate is shown by a percentage number. Here is an example of *ping* output which sends 5 *echo request* probes with the 5 second timeout. as we can see in each line the RTT value shown for each particular packet.

*Ping* utility output example.

```
ping -I 129.240.66.74 193.10.227.85 -c 5 -w 5
PING 193.10.227.85 (193.10.227.85) from 129.240.66.74 : 56(84) bytes of data.
64 bytes from 193.10.227.85: icmp_req=1 ttl=53 time=13.2 ms
64 bytes from 193.10.227.85: icmp_req=2 ttl=53 time=13.2 ms
64 bytes from 193.10.227.85: icmp_req=3 ttl=53 time=13.5 ms
64 bytes from 193.10.227.85: icmp_req=4 ttl=53 time=22.0 ms
64 bytes from 193.10.227.85: icmp_req=5 ttl=53 time=13.4 ms


--- 193.10.227.85 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 13.254/15.128/22.051/3.464 ms
```

It is possible to specify the number of *echo request* packet when running *ping* as an option otherwise ping starts sending packets until a quit signal stop that. When a ping stopped due to the specified number of *echo request* packets have been sent or with a quit signal, *ping* illustrates a statistical report. *Ping* report includes the number of request probes have been sent, number of reply packets have been received and the percentage of the packet loss. In addition a statistical summery of minimum, average, maximum and standard deviation of the RTTs is exposed.

#### 2.3.2.4 *Ping* Issues

- Loss asymmetry: Packet loss in network have a significant impact on application performance specially for *Transmission Control Protocol (TCP)* [65]. As have been shown in [53] with the moderate packet loss in the network the TCP bandwidth is fits to $1/\sqrt{lossrate}$ .

  *Ping* sends a probe from source S to the destination D and waiting for the response packet within a time period, hence measures the packet loss rate by calculating the not responded probes. Since the packet loss rate mostly different from forwarding direction and the reverse direction leads into the *loss asymmetry*. *Ping* gives no information about whether probe packet was lost or the response packet was lost which is one of the *ping* problems in reporting the packet loss rate. If we consider the loss rate in forwarding path as $loss_{fwd}$ and the loss rate in reverse direction as $loss_{rev}$ the loss rate is:

  $$1 - ((1 - loss_{fwd}).(1 - loss_{rev}))$$

  The *loss asymmetry* is important in some applications or protocol, for instance in streaming the packet loss in opposite direction have no or little impact on performance of streaming. In TCP also the loing acknowledgement packet is tolerated better than the losing data packet [71]

- ICMP filtering: Due to the abusing ICMP services by attackers and it's destructive impacts on the network functionality and performance, some networks or operating systems (e.g. Solaris) limited the ICMP response or filtered the all ICMP traffics. In addition some firewalls and load balancers responding on behalf destination host. Since *ping* is an ICMP-based utility, these kind of restriction of ICMP by firewalls and networks result into the *ping* efficiency reduction.

### 2.3.3 Traceroute

*Traceroute* is a networking tool and it is an essential tool for network administrator which is written by Van Jacobson [78] and mostly used as network debugging tool. The *traceroute* utility is used for discovering the Internet path and transit delay time of packet which travelling to the destination across the Internet Protocol (IP) [8]. *Traceroute* is widely used for diagnosing network problems by network operator such as routing failure or misconfiguration and poor network performance. In addition is used for evaluating the performance of ISPs and Internet mapping [41].

#### 2.3.3.1 Functionality of Time To Live field in *traceroute*

Internet Protocol (IP) have a *Time To Live* (TTL) field in the header [66] while the aim of this field is to prevent a packet infinitely travailing around a routing loops that cause network problems. The TTL filed is 8 bits in the IP header and elucidate the time in second until the packet must discarded, thus the maximum value can be 255. TTL value is decremented by 1 per Internet router thus when a router face the TTL zero should discard the packet and not forward the packet. In this way an Internet Control Message Protocol (ICMP) [14, 63] packet will send to the sender in order to made sender inform that the packet has expired lifetime. In the reason of network performance all Internet router decrement the TTL by 1 per hop without considering the processing time.

*Traceroute* is supported for both IPv4 (*traceroute*) and IPv6 (*traceroute6*) [52].

#### 2.3.3.2 How *traceroute* works

*Traceroute* works by manipulating the TTL field of the header. In order to exploring the path from source S to a remote host D, *traceroute* by default builds a UDP [64] packet with the TTL 1, and IP address of D as destination. When a packet enters an Internet router, the TTL value will decremented by 1, therefore at the first hop the TTL value become zero and the packet can not go further in depth. In this way the hop 1 sends an *ICMP Time Exceeded* message to the sender with specifying the IP address of itself in the header in order to *traceroute* identify the hop 1 address in the path to the destination R. *Traceroute* proceeding in this fashion by increasing the TTL value one by one, receiving the *ICMP Time Exceeded*, and building a list

of routes until reaching the destination. The sender identifies the destination has been reached by receiving an *ICMP port unreachable*. In Unix based systems by default *traceroute* sends UDP probes to the high-numbered destination ports and increment the destination port number per probes in order to matching the responses to the probes. The UDP base probes starts with 33435 destination port number and increments the the destination port number for each probe. The possibility of probing an open service on a machine is diminish significantly within choosing the high-number of destination ports by *traceroute* [48]. *Traceroute* by default sends three probes for each hop and each probe waiting until *traceroute* receive a reply for the previous probe in addition *traceroute* waits 5 seconds for reply before concluding there will be no reply for that probe. Here there is an example of *tracroute* output which tracing the path from Simula research laboratory in Norway to University of Hainan in China.

*Traceroute* utility output example.

```
traceroute to 113.59.104.58 (113.59.104.58), 30 hops max, 60 byte packets
 1  158.39.4.1 (158.39.4.1)  1.455 ms  1.386 ms  1.590 ms
 2  128.39.37.1 (128.39.37.1)  1.206 ms  1.263 ms  1.262 ms
 3  simula-gw.uio.no (128.39.70.33)  1.594 ms  1.938 ms  2.526 ms
 4  oslo-gw7.uninett.no (158.36.84.149)  1.545 ms  1.824 ms  1.695 ms
 5  ifi2-gw.uninett.no (128.39.254.185)  2.510 ms  2.502 ms  2.491 ms
 6  stolav-gw2.uninett.no (128.39.254.98)  2.249 ms stolav-gw2.uninett.no
 (128.39.254.38)  2.214 ms  2.139 ms
 7  dk-uni.nordu.net (109.105.102.25)  37.366 ms  37.450 ms  37.438 ms
 8  dk-ore.nordu.net (109.105.97.13)  29.395 ms  29.432 ms  29.417 ms
 9  kbn-b4-link.telia.net (62.115.11.77)  55.129 ms  55.231 ms  55.222 ms
10  kbn-bb2-link.telia.net (80.91.247.244)  29.336 ms  29.127
ms kbn-bb1-link.telia.net (80.91.246.46)  33.342 ms
11  nyk-bb1-link.telia.net (213.155.134.50)  110.652 ms nyk-bb2-link.telia.net
(80.91.254.91)  184.207 ms nyk-bb1-link.telia.net (213.155.134.50)  110.769 ms
12  sjo-bb1-link.telia.net (80.91.245.96)  194.518 ms sjo-bb1-link.telia.net
(213.155.133.239)  196.274 ms  196.163 ms
13  chinaunicom-ic-127288-sjo-bb1.c.telia.net (213.248.73.190)  195.012 ms
195.863 ms chinaunicom-ic-141282-sjo-bb1.c.telia.net (213.248.71.90)  183.790 ms
14  219.158.30.49 (219.158.30.49)  400.935 ms  407.275 ms  400.919 ms
15  219.158.97.17 (219.158.97.17)  402.712 ms  403.318 ms  397.896 ms
16  219.158.11.37 (219.158.11.37)  468.431 ms  468.175 ms  462.123 ms
17  219.158.24.218 (219.158.24.218)  446.403 ms  453.677 ms  447.325 ms
18  221.11.154.182 (221.11.154.182)  453.477 ms  453.527 ms  459.166 ms
19  * * 221.11.154.202 (221.11.154.202)  485.155 ms
20  113.59.104.58 (113.59.104.58)  494.084 ms  494.028 ms  460.933 ms
```

As we can from first line see the default maximum number of hops for *traceroute* is 30 hops which it can be set optionally by option "-m" while the maximum number can be 255. *Taceroute* reported the number of hops, IP address of each hop router, and the reply time in millisecond. The "*" in hop 19 shows the *traceroute* or ICMP message was dropped by network. [52, 59]

#### 2.3.3.3 *Traceroute* limitations

*Traceroute* is a dominant utility for discovering the Internet path while have experience several limitations.

1. *Traceroute* just able to see what the Internet accept to expose and is a ***routing dependent*** tool. For instance *traceroute* is not able to traverse backup links.[18]

2. When we tracing a route from Source **"S"** to Destination **"D"**, *traceroute* is be able to reveal the path from S to D and can to elicit route fro D to S which is likely different. This issue is known as ***unidirectional*** [19, 59].

3. As we can see from the *traceroute* output example in line 11 that there are more than one IP address has been shown. The first prob reached IP address *213.155.134.50* while the second hop 11 prob meet IP address *80.91.254.91*. This kind of routing can happen due to the ***load balancing***. *Traceroute* is sensitive to the Internet path load balancing which is widely used by ISPs in order to increase the availability and reliability of their networks. Existence of multiple path which is inferred from essence load balancing will be led into the exploring *false link* by *traceroute*. As illustrated in Figure 2.8 the probe with TTL = 2 discovered router $R_2$, and the next prob with TTL = 3 reached router $R_4$, Hence a false link between router $R_2$ and router $R_4$ will deduced [18, 19, 59].

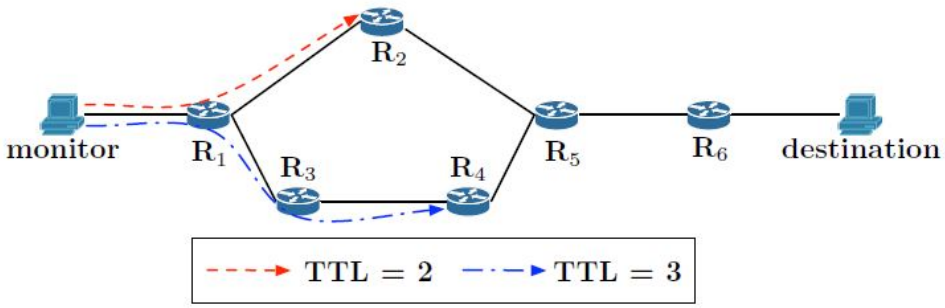



Figure 2.8: Impact of load balancing on *traceroute* result.[18].

4. Exploring repeatedly the same interface by *traceroute* led into the huge network consumption. This issue known as ***redundancy*** that happen when a large number of *traceroute* probes emerged from a large number of monitoring machine and meet a selected destination which seems as a distributed denial-of-service (DDoS) attack. There are two different form of redundancy 1)*intera-monitor redundancy* occurs when all *traceroutes* produce from a single point and 2)*inter-monitor redundancy* happens when *traceroutes* generate from several monitor machine to the same interface [20, 21].

5. *Traceroute* can not explore the ***hidden routers*** such as *Multiprotocol Label Switching* (MPLS) tunnels. *Traceroute* probes which traversing MPLS tunnels can not show tunnel content since MPLS tunnels adumbrate the topology information [18]

6. *Traceoute* extract out the paths just in the IP `network layer` and is not able to reveal the `Link layer` routes such as point-to-point or Ethernet links [59].

7. some Internet routers are configured to answer a limit ICMP message for instance limitation of one ICMP per second in order to preserving network bandwidth and router resource resource [59]. On the other hand some routers are not replying to the *traceroute* probes at all which these router known as ***Anonymous Routers*** which result in a hole in the derived path. The "*" in line 19 from example above is the result of *traceroute* when meets these routers [18].

## 2.4 Related Work

### 2.4.1 Hidden Router

*Multiprotocol Label Switching* (MPLS) is used in order to reduce time of forwarding decision making in IP routers [70]. MPLS networks implemented on the IP routers which use a label distribution protocol [1]. MPLS deployed by adding 32-bit *label stack entries* (LSE) in front of IP header in IP routers. The following MPLS *label switching routers* (LSRs) in the network use the LSE field which determined the forwarding actions. At MPLS hop, packet forwarded by the exact match in 20-bit label of LSE by replacing the incoming packet label by the corresponding outgoing label founded in switching table, therefore the MPLS forwarding engine is much more lighter than the IP forwarding engine because finding a match for a label is more easier compare to the long IP address prefix.

In addition MPLS LSE has time-to-live field (LSE-TTL) and type-of-service field hence when LSE-TTL expires there is a probability that MPLS router send ICMP time-exceeded message reply. Therefore the ICMP extension object [7] came as a debugging way for this malfunctioning since it allows LSR to insert a MPLS label stack to the ICMP time exceeded message and since 1999 this extension have implemented by the router manufacturers.

On the other hand if the first MPLS router (the *ingress* Label Edge Router-LER) in the *Label Switched Path* (LSP) copy the IP-TTL value in the LSE-TTL field instead of setting it to an arbitrary value, then the following LSRs in the LSP will reveal themselves even by ICMP message even whether they did not implement the ICMP extension object. The *ttl-propagation* option in the router is used to configure this action.

The *"ttl-propagation"* and *"ICMP extension"* are two technologies as MPLS transparency technology. But unluckily these two technology have not been implemented universally, hence the traceroute-based utilities can not reveal the MPLS deployment and false router-level link can be concluded.

Donnet et al. illustrated that there are four different groups of MPLS tunnels according to the these two MPLS transparency techniques. Figure 2.9 shows these four classes [22].

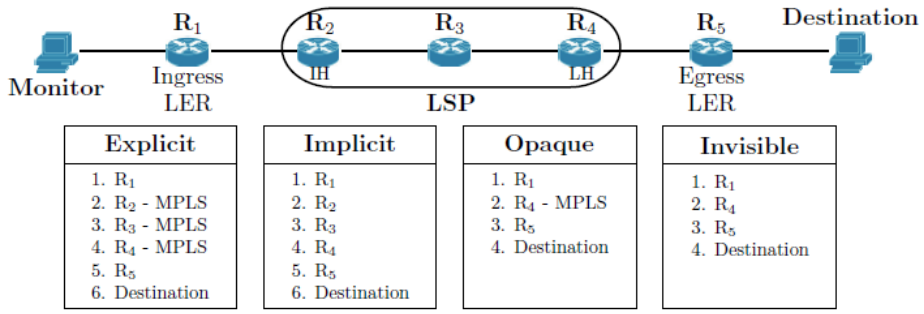



Figure 2.9: MPLS tunnel configurations and corresponding traceroute behaviours [22].

- *Explicit tunnel*: The both *ttl-propagation* and *ICMP extension* technologies are implemented. The internal structure of tunnel is visible and each LSR in LSP reveal themselves by MPLS flag.
- *Implicit tunnel*: The *ingress*-LER enabled the *ttl-propagation* but the *ICMP extension* is not implemented in LSRs. Therefore the internal structure of tunnel is visible but the existence of MPLS is not revealed.
- *Opaque tunnel*: The LSRs implement the *ICMP extensions* but the *ingress*-LER does not enable the *ttl-propagation*. The Internal structure is invisible and just the LH router that pops the MPLS label reveal LSE. As can be seen from figure 2.9 the false link between $R_1$ and $R4$ is inferred.
- *Invisible Tunnel*: The *ingress*-LER does not enable the *ttl-propagation* and also the LSRs do not implement the *ICMP extension*. The internal structure is invisible and non of LSRs flag themselves as MPLS. As can be see from figure 2.9 this kind of tunnels cause false link (link between $R_1$ and $R_4$). Up til now there is no techniques to discovering these types of tunnels.

Donnet et al. proposed that at least 30% of the traceroute reveal *explicit* MPLS tunnel and more than 5% of collected IP interface exhibit MPLS capability. They also suggested that the *implicit* tunnel is three times less common than the explicit one and the *opaque* tunnels are very infrequent. Donnet et al. developed a finger print mechanism in order to revealing *implicit* tunnel that hide the use of MPLS and a technique in order to show the length of *opaque* tunnel. They estimated that the *invisible* tunnel are 40 to 50 times less frequent than the *explicit* tunnel.

### 2.4.2 Load balancing

Internet path *load balancer* is used by Internet Service Providers ISPs and multihomed ISPs in order to increase the reliability and availability of their networks and services. *Load balancer* is a router in the network that performing load balancing [18]. *Traceroute* which is the dominant path discovery utility have some deficiencies in facing a path *load balancer*. There are three kinds of load balancing [12, 13]:

- *Per flow* that each packet is assigned to a flow according to the packet header. The flow identifier can be the *five-tuple* fields in the packet header which includes *Source Address, Destination Address, Protocol, Source Port, Destination Port*. The *Type of Service (Tos), ICMP field* and *checksum* fields also can be used as flow identifier. hence the *load balancer* router forwarding the packets which belongs to a flow to the same interface.
- *per packet* attempts to keeping a balance the load on each link with out considering the packet header.
- *per destination* is the same as the the classic routing that routes packet only based on the destination, regardless of the packet source.

The existence of the *Load balancer* in a path implies more than one active route in a given source to destination. *Traceroute* suffers from the presence of *load balancer*. The classic *traceroute* sends probes by different UDP destination port number different ICMP checksum value which cause the *per-flow* load balancer distinguish them as different flows and forwarding them to different path. Therefore as we can see from figure 2.10 this behaviour make the the classic *traceroute* solicit response from the unconnected routers which is known as *false link*.

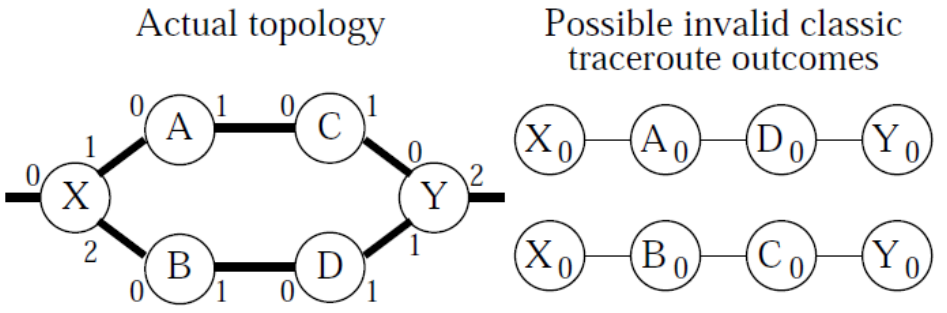



Figure 2.10: Load balancing can cause false inference of router-level links [50].

Augustin et al. developed a new *traceroute* called *paris-traceroute* for networks with *load balancers* and with the purpose of avoiding false link inference from classic *traceroute* [5]. The innovation key of *paris-traceroute*

is to controlling the probe header fields in the way that with the presence of *per-flow* load balancer, probes toward a destination follow the same link. *Paris-traceroute* is also gives information about the presence of *per-packet* load balancing but can not perfectly enumerate all possible paths due to the random nature of *per-packet* load balancer.

*Paris-traceroute* customizes the probe packet in order to make consistent result across the *flow-based load balancer*. *Paris-traceroute* cannot alter the first four octets and plus *Type of Service (ToS)* field of transport layer header because they are used by load balancers, thus *paris-traceroute* varies the seventeenth and eighteenth octets of packet header. For the UDP probes *paris-traceroute* alters the *checksum* field, But to avoid producing malformed probes it needs to also modify the packet payload.

Like the classic *traceroute*, *paris-traceroute* uses the *sequence number* field for echo ICMP probe. But over the counter of classic *traceroute*, *paris-traceroute* keep the *checksum* constant by changing the ICMP identifier. *Paris-traceroute* also sends TCP probes like Toren's variant *tcptraceroute* [51] and uses the *Sequence Number* field in order to make the first four octets heade constant.

*Paris-traceroute* found that the classic *traceroute* traces are artifacts which extract from the side-by-side measurement of classic *traceroute* which prevented by *paris-traceroute*. *Paris-traceroute* can discover the accurate paths because the classic *traceroute* is incapable to illustrate real path under the *per-flow* load balancer. Finally by use of *Multipath Detection Algorithm (MDA)*, *paris-traceroute* can measure the path under *per-destination, per-flow* and *per-packet* load balancing. It exposed that the 39% of source-destination pairs traverse a *per-flow* load balancer and 70% traverse a *per-destination* load balancer [3, 4, 77].

Luckie et al. evaluated the inaccuracies of routes caused from the false links [50]. For the evaluation of the data collection from the classic *traceroute* they used two model of probing: 1) Macroscopic probing in the fashion *Archipelago* [10] which is a macroscopic Internet topology mapping project that tracerouting toward millions of destination. 2)ISP probing in the fashion of *Rocketfuel* [73] which is an ISP-mapping traceroute-based tool in order to discover the map of individual ISPs. According to the macroscopic topology data collection they used both capturing and not capturing load balancing phenomena technique. Afterwards they used the alias resolution technique in order to conclude whether the classic *traceroute* cause a false router-level link. They collect the data towards 365k destination and they saw that 2.71% of routes in UDP graphs and 0.76% of routes in ICMP graphs falsely inferred in the reason of presence of load balancer. Therefore the error of macroscopic topology data with ICMP method was minor which means the most *per-flow* load balancer does not cause false link in macroscopic topology derived by classic *traceroute*.

According to the ISP probing they found that the impact of using classic *traceroute* is more significant in a way that two third of collected links towards the targeted ISPs were suspicious and the false link inferred from classic *traceroute* caused router degree inflation and ISP path diversity in the ISP network.

### 2.4.3 Unidirectionality

*Traceroute* discover a sequence of routers from source to destination with the round-trip delays of each hop while suffers from lack of *reverse* path information. It means the classic *traceroute* cannot expose the route from the destination to the traceroute source which is known as *unidirectionality*. Since due to the traffic engineering and routing policy the path are asymmetric means the *forward path* ($S \rightarrow D$) is different from *reverse path* ($D \rightarrow S$) which is a phenomena in Internet modeling [34]. As illustrated in figure 2.11 the forward probe traversed $AS_1$, $AS_4$, $AS_2$, $AS_3$ in order to reach the destination. But the reverse probe go through $AS_3$, $AS_6$ and $AS_1$ to reach the source.

Lacking knowledge about the reverse path is an obstacle for both operators and researchers for more precisely network troubleshooting and Internet mapping. Despite there are several public web-accessible traceroute servers around the world, but their numbers are limited and they are not designed for receiving a heavy load and regular monitoring.

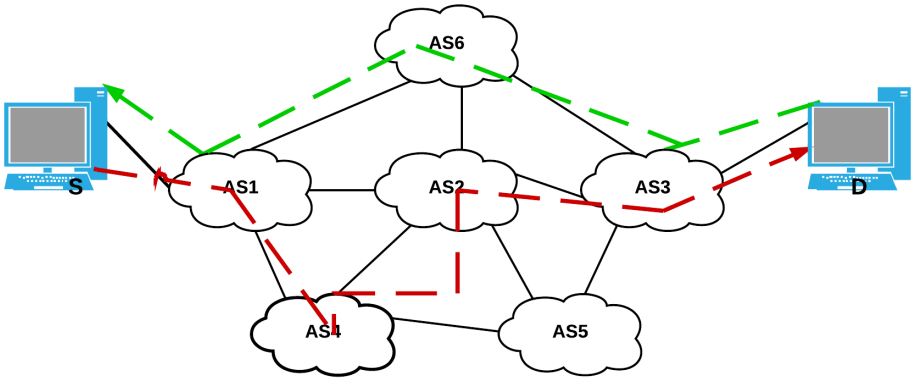



Figure 2.11: Path asymmetry and difference of forward path and reverse path.

Katz-Bassett et al. proposed the ***reverse traceroute*** tool in order to solve the reverse path invisibility of the classic traceroute [41]. The *reverse traceroute* tool is able to expose the fundamental information like classic traceroute such as IP address and round-trip-time of each hop but in the reverse direction from the destination to the source and also it works in the case that the destination is not under control. *Reverse traceroute*

tool just need a destination that responds to the probes like the classic traceroute does not need more requirements than the classic traceroute. *Reverse traceroute* uses several controlled vantage point and three measurement mechanism and builds the reverse path incrementally. It starts by measuring the path from the vantage point to the source while the Tree-like structure of routes is considered in order to avoiding probing redundancy. The reverse path is built by stitching the observed hops together incrementally hop-by-hop which the destination hop comes at first of path.

*Reverse traceroute* uses three different measurement mechanism to build backwards.

- First, since the the Internet routing is destination-based the *reverse traceroute* can capture and put together one hop at a time.
- Second, *reverse traceroute* use the *timestamps* and *record route* options of IP header in order to identify the hops along the reverse path.
- Third, it uses limited IP spoofing, spoofing from one vantage point to another in order to overcome the shortage of IP timestamps option.

*Reverse route* uses the combination of these three techniques to measure the reverse path from an arbitrary destination to the user. The *reverse traceroute* tool can find in a median case *87%* hops which is observed in a direct traceroute measurement along the same path.

Katz-Bassett et al. demonstrated that *reverse traceroute* tool can discover more than thousand of peer-to-peer invisible links to both BGP route collector and traceroute-based mapping. It can also measures the latency of backbone links.

# Chapter 3

# Approach

This chapter gives an overview about the goal and difficulties of this project, measurement environment, measurement strategy, data collection strategy, experiments and analysis procedures which the detail of the implementation is in following sections. The focus of this research as noted in the problem statement is to measuring the stability of Internet paths over IPv6 and comparing with IPv4 paths.

Internet path means the sequence of intermediate router that a packet follow from a source node till reaching a destination node. The path stability is meaning invariable or constant path between a source-destination pair in a unit of time. The observation route changes of IPv6 paths vs. IPv4 paths and analysis the differences in stability is this research goal. In order to achieve this knowledge this research can break up into following parts:

- **Measurement environment setup**

  These kind of measurement and research can not implemented on a production network as the result of influence of measurements on performance and functionality of the network and in addition will have effects on the measurement data that are collected due to the production network policies and their firewall configurations which cause disordering and finally produces inaccurate results. Therefore in order to have reliable and accurate data a testbed is needed which can provide the realistic, flexible and capable infrastructure with high performance for executing test and measurements.

- **Measurement and data collecting** There are several tool in order to measuring the Internet paths which some of them has been explained in the previous section. Since characterising the stability of Internet path requires active, long-term and continuous Internet routes collection, a measurement tool is needed to built up by using the current Internet path measurement utilities in a way that can collect the route data in set of IPv4 and IPv6 paths and running continuously. The measurement experiments, data collection process, types of data to be collected and data storing process would be defined here.

- **stability comparison and analysis** After obtaining Internet paths in a period of time, An investigation process on the recoded paths is needed in order to recognition of paths changes over a period of time. We develop metrics and data processing procedures in order to characterising path stability as well as providing a comparison between IPv4 and IPv6 paths.

## 3.1 Testbed Infrastructure

The first step in this project is to defining a testbed which provides reliable and capable infrastructure with Internet connectivity over IPv4 and IPv6.

In this research we use the *NorNet Core testbed* infrastructure which is a multi-homed testbed distributed all over Norway and some other countries [32]. The multi-homing of NorNet testbed means that each site is connected to more than one Internet Service Providers (ISPs) which makes NorNet more available and reliable. Both Internet protocols (IPv4 and IPv6) are available for most of service providers on NorNet testbed, hence IPv6 is reachable on most of sites. The detailed characteristics of NorNet testbed is provided on background section 2.2.

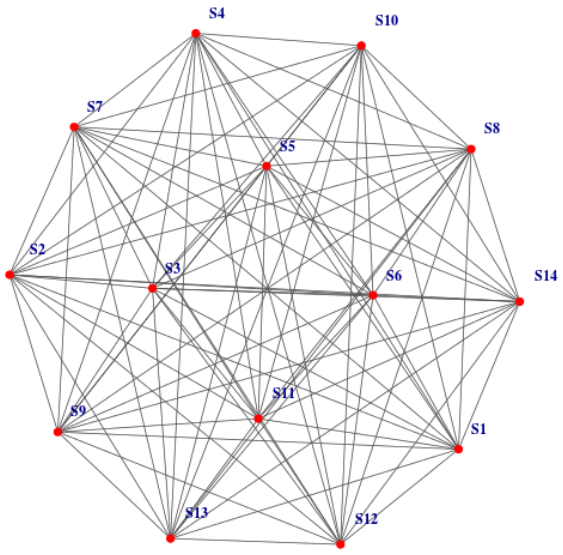



Figure 3.1: Full connection of measurement environment.

At this moment the NorNet Core have *14* sites which 11 sites are located all over Norway, 1 site is in Essen-Germany , 1 site is Karlstad-Sweden, And 1 is in Hanian-china. The connection of each site to other sites can built a full mesh of connection as has been shown in figure 3.1.

Each Sites in NorNet Core connected to more than one ISPs, hence the possible connection between two sites in NorNet testbed is not just one simple connection, but also the connections from all ISPs of source site to all ISPs of destination site are possible. The complexity of possible connections between each site are not end up here due to the different Internet protocols (IPv4 and IPv6) that is available from the most of NorNet Core Service Providers. The example of possible connection between two sites in NorNet Core testbed, while one site connected to three ISPs and the other site connected to two ISPs, and all ISPs also provide connections over IPv6 has been shown in figure 3.2.

In figure 3.2 the blue line are connections Over IPv4 and red line are connections over IPv6.

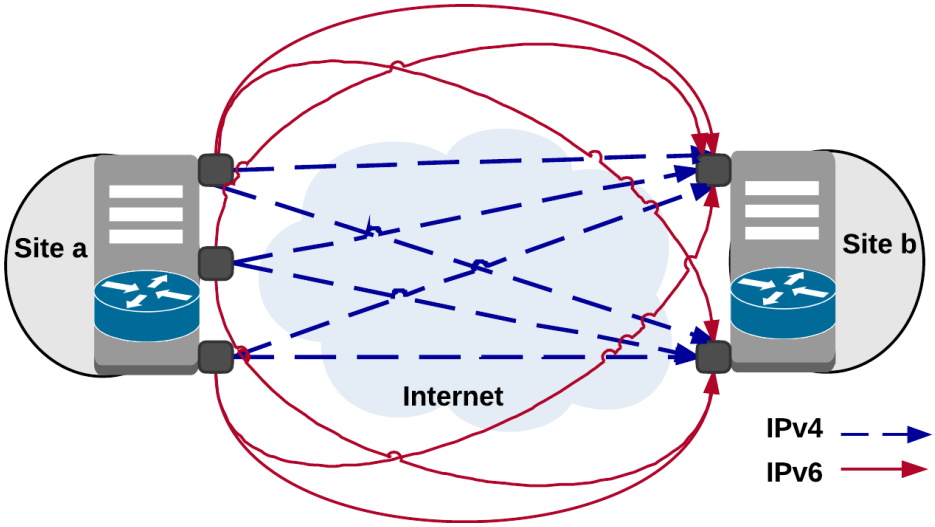



Figure 3.2: Possible connection between 2 sites via IPv4 & IPv6 in NorNet testbed.

Here If we consider:

$$\begin{cases} S_a \rightarrow P_{a1}, P_{a2}, P_{a3} \\ S_b \rightarrow P_{b1}, P_{b2} \\ |P_{Sa}| \times |P_{Sb}| = 3 \times 2 \end{cases} \tag{3.1}$$

And If IPv6 is available from all ISPs in both sites $S_a$ and $S_b$, the number of connection is:

$$|P_{Sa-v4}| + |P_{Sb-v6}| = 6 + 6 = 12$$

Generally If we consider $P_i \in [1, 255]$ as *NorNet provider Index*, and $TP4_n$ while $n \in [sites]$, as the *Total Service Providers* in each site which provide connection over IPv4. For instance the $TP4_a = 3$ and $TP4_b = 2$. The

possible connection over IPv4 from the a source site with the total available ISPs of $TP4_s, s = source$ is following this formula:

$$C_4 = TP4_s \times \{(\sum_{n=sites} TP4_n) - TP4_s\} \tag{3.2}$$

Formula 3.2 shows the possible connection from one site as the source to the other sites of NorNet testbed just over IPv4. While The possible connections from one site to all other sites over IPv6 , Over both IPv4 and IPv6, and total possible connection from all site to all other sites which create a mesh of connections are as the formula (1), (2), and (3) respectively.

$$C_6 = TP6_s \times \{(\sum_{n=sites} TP6_n) - TP6_s\} \tag{3.3}$$

$$C_4 + C_6 = \big(TP4_s \times \{(\sum_{n=sites} TP4_n) - TP4_s\}\big) + \big(TP6_s \times \{(\sum_{n=sites} TP6_n) - TP6_s\}\big) \tag{3.4}$$

$$\sum_{sites} C_4 + \sum_{sites} C_6 \tag{3.5}$$

In NorNet Core provide connection between each sites by Network Layer tunneling, while each combination of source-destination pairs based on Internet protocols and Service Providers are separately tunneled. Since in this project we are capturing the intermediate routers between source and destination we are using the public network connection between the sites instead of tunnels while the tunnels can not see the intermediate routers.

## 3.2 Measurement Design

This section describes an approach for measuring the Internet paths over IPv4 and IPv6 in the fashion of running continuously and actively recording the paths for a long time as well as storing data collected in a way that can be used for the future analysis and discovering the path changes. Hence this measurement could be split into three subsections:

### 3.2.1 Internet Path measurements

In order to capture the Internet path it is needed to sending a probe packet from a source node to a destination node and capturing the intermediate routers that the packet traversed through them. As has been mentioned in the previous part that we are using the NorNet Core infrastructure which is providing multi-connectivity from each site to the other sites in addition to the connections over IPv4 and IPv6, therefore the measurement tool have to be capable to obtaining the routes of all combination from the source site to the all destination sites. It means that taking the combination of all local ISPs to all ISPs of each remote site over both for IPv4 and IPv6 which

is following the formula 3.4 in the previous part. The routes from each source-destination pairs, and both IPv4 and IPv6 should be distinctly captured which provide is more than hundred route.

- On the other, the route that a packet travels from a source to a particular destination is mostly different from reverse path from that destination to the source of packet and also due to the *unidirectionality* of current Internet path measurement utilities such as *traceroute* and *ping* which have been discussed in the sections 2.3.3.3 and 2.4.3, it is required that the measurement also runs from all sites and traces the paths to all other sites since brings a mesh of recorded pat that following formula 3.5.

- On top of those in order to discover the stability or instability of the Internet paths, it is necessary to run the measurement tool continuously for a long time and recording all the routes for future investigation. Therefore this tool is needed to run as a service and actively capturing the all the routes from all sites to all other sites over IPv4 and IPv6.

- The measurement tool should extract other valuable types of data in addition to the sequence of intermediate routers between each combination of source-destination pairs and record them which can be useful in the future path changes discovery and future analysis.

### 3.2.2 Data collection and storing

As the measurement service is running and capture valuable data from each combination of source-destination pairs, it is needed that each collected data stored in a way that can be reused for the future analysis. The process of data storing could be divided into two parts:

#### 3.2.2.1 Local Storing

As the measurement is running, the collected data must be stored on the hard disk temporary. The data could be stored in the format of files which is separated by each source-destination pairs over IPv4 and IPv6 combination. It means that in a single run of measurement service from each site, the Internet route from current site to all remote sites with all ISPs and Internet protocols would be collected and stored in separated files. Hence the number of stored files in each iteration of service is the same as number of connection from local site to the remote sites which follows formula 3.4.

#### 3.2.2.2 Remotely Storing

In the reason of storing collected data in the format of file locally in each sites, the files are accessible only from sites and just route data from current site to other sites are reachable. Therefore in order to ease the future analysis, a process of centralizing the collected data is needed in a way that all routes records can be accessible from a single point. The remote storage could be a database on a physical or virtual machine which all sites of NorNet Core could have access to that database and insert collected data to the database. choosing database as the central dataset make the process of path changes identifier and analysis part feasible.

The central dataset could be a table in the database with identifiers that can separate each source-destination pairs for the different iteration of Internet route measurement which can easily distinguish for the future analysis.

The route collected data could be transfer to the central storage directly with out storing locally in each site, but it will increase the network I/O load, hence the route data would stored locally at each site as file, then process of transferring data to the remote storage would be run from time to time the.

## 3.3 Analysis Design

The next step after running the measurement and having the Internet path data of different source-destination pairs over IPv4 and IPv6 for a period of time in the form of a complete dataset, is to extract route changes for each source-destination pairs and comparing IPv6 path changes with IPv4 path changes. We use the periodic path probes to identify path changes, so that we can analyse their frequency and type.

In this way the analysis can be split into the following components:

### 3.3.1 Path change classification

We categorise path changes in three different classes, based on the observed old and new path:

#### 3.3.1.1 Hop count changes

Hop count is the number of intermediate routers between source node and destination node. If we consider the $H_C$ as the hop count of an Internet route which can be equal to $n \in [1, 255]$, then the hop count change is the changes in $H_C$ of that route that can change into more hops or less hops . Hence from this time on we can consider the changes in Hop count as $C_{HC}$

as bellow:

$$C_{HC} \rightarrow \begin{cases} H_{C-old} > H_{C-new} \\ H_{C-old} < H_{C-new} \end{cases} \tag{3.6}$$

$$H_C = n \in [1, 255]$$

#### 3.3.1.2 IP changes

Internet path consist of a sequence of intermediate router's IP addresses. For instance if a route between source to destination consist of **n** number of hops, therefore that path is constructed by the sequence of $P_i$ address which $i \in [1, n]$ is the index of IP address in the path. The IP changes which from this time on we consider it as $C_{IP}$, is the changes in the IP of a particular index in that path. It can be clear from following equation:

$$C_{IP} = \begin{cases} P_{i-old} = \alpha \rightarrow Pi - new = \beta \\ \alpha <> \beta \end{cases} \tag{3.7}$$

$$i \in [1, n]$$
$$n = Number \quad of \quad hops \quad in \quad path \quad P$$

#### 3.3.1.3 Star changes

Fetching Internet path by current Internet path measurement utilities such as *traceroute* or *ping* suffers from some drawbacks which can not show the IP address of the *Autonomous Router* in the path which has been discussed in part **7** of 2.3.3.3. Hence it is possible to see some IP address in the path changes into the **'*'** or we can not see the IP address of some routers at all. The Star changes that we consider it as $C_S$ is the changes of IP address of particular index in the path to star or changes from star to IP address. If we consider $P_i, i \in [1, n]$ as the IP address or router identifier in the path which is between 1 and **n** = number of hops, the star changes is as follow:

$$C_S = \begin{cases} P_i = \alpha \rightarrow P_i = * \\ P_i = * \rightarrow P_i = \alpha \end{cases} \tag{3.8}$$

$$i \in [1, n]$$
$$\alpha = IP \quad address$$
$$n = Number \quad of \quad hops \quad in \quad path \quad P$$

### 3.3.2 Grouping path changes

After classifying the path changes and identifying different source of path changes as discussed in the previous part, and splitting them up into $C_{HC}, C_{IP}, C_S$, which are *hop count changes*, *IP changes* and *Star changes* respectively. It is needed to sorting them into an event and grouping the events in order to identifying the types of path changes. Creating an event out of the path changes would accomplish by introduce a **threshold** which is a time period. If there is a time period without no changes equal or longer than the *threshold* will be led into two separate events. This data processing will create a distinction between Internet path that had frequent and continuous changes, and the Internet paths that experienced less or rarely changes. Our goal is to observing the temporal aspects of path changes. whether they happen evenly spread out in time, or do they come in bursts? This is important for assessing path stability. In this fashion an events $E_i, i \in [0, n]$ in a path could have several values as follow:

$$E_i \rightarrow \begin{cases} Size = Numberofpathchanges. \\ Duration = (Event - end - time - Event - start - time) \\ Type \rightarrow \begin{cases} Numberofhopsbeforeevent. \\ Numberofhopsafterevents. \end{cases} \end{cases} \tag{3.9}$$

$$i \in [1, n]$$

$$n = Number \quad of \quad events \quad in \quad a \quad single \quad path$$

### 3.3.3 Average of path changes per day

Another analysis that could be run is to getting the total number of path changes of a single path as $TC_i$ , $i \rightarrow \{All \quad paths \quad combination\}$ for the whole time period of data collecting, and calculating the the average as $A_i$ , $i \rightarrow \{All \quad path \quad combinations\}$ of path changes by dividing Total number of path changes to the number of days of the whole time period of data collection as $d_i$, which follows following formula:

$$A_i = TC_i / d_i \tag{3.10}$$

$$i \rightarrow \{All \quad paths \quad combination\}$$

The Average value needed to calculate for each source-destination and over IPv4 and IPv6 distinctly and also for all different kinds of path changes (*hop count, IP and star change*). The overall overview could be proposed by one or several plot, and the average stability and instability of Internet path over IPv4 and IPv6 would be concluded from that graph.

### 3.3.4 Distribution of path changes

An other data processing procedures that could be done is to proposing how the path changes are distributed over the change days. It could be

calculated by getting the total number of days that path changes have been seen in a single source-destination pair as $TD_i$, while $i \rightarrow \{All \quad source - destination \quad pairs\}$, as well as total number of path changes of the selected path as $TC_i$ in a total time period of data collection and we can consider as $d_i$. Proposing the distribution of path changes over time could be done by showing the potential associations between *fraction of days* that had changes and *average number of path changes per day* and could be calculate by:

$$\begin{cases} Fraction \quad of \quad days_i = TD_i/d_i \\ Fraction \quad of \quad path \quad changes_i = TC_i/TD_i \\ d_i \rightarrow \{Total \quad days \quad of \quad data \quad collection \quad of \quad source - destinationpair \quad i\} \\ TD_i \rightarrow \{Total \quad path \quad change \quad days \quad of \quad source - destinationpair \quad i\} \\ TC_i \rightarrow \{Total \quad path \quad change \quad of \quad source - destinationpair \quad i\} \\ i \rightarrow \{All \quad source - destination \quad pairs\} \end{cases} \tag{3.11}$$

By illustrating the relation between the *fraction of days* that we had changes and *fraction of path changes of those days*, it is possible to figure out, how path changes are distributed over the change days. In addition comparison of IPv4 and IPv6 Internet path changes distribution over time would be concluded from the graph.

### 3.3.5 Distribution of Path length

One data processing that could be consider is the distribution of Internet path length over IPv4 and IPv6 by taking the length of Path as $L_i$ , $i \rightarrow \{All \quad path \quad combinations\}$, which is the number of intermediate hops in a single path **"i"**. This comparison would be proposed in graph and comparing IPv4 and IPv6 Internet path length and would be illustrate whether IPv4 have longer Internet path in average or IPv6.

In this research using the statistics is needed in order to calculate the stability or instability of Internet path and comparing the IPv6 Internet path with IPv4 Internet paths as the expected conclusion. In addition converting the results out of this data processing into graphs could ease the process of analysis and make this research meets the goal which is *measuring and comparing the stability and instability of Internet path over IPv4 and IPv6*.

## 3.4 Operationalization plan

As have been noted in the previous section the measurement should be perform in each site of NorNet Core testbed separately and the the collected data would be centralize. Hence the operational plan in each site of NorNet Core testbed is proposed in figure 3.3.

The operational function would be start from measurement service on figure 3.3 in each site and would be act as follow:

1. **Measurement service**:
   The *measurement service* is a tool which could be written with one

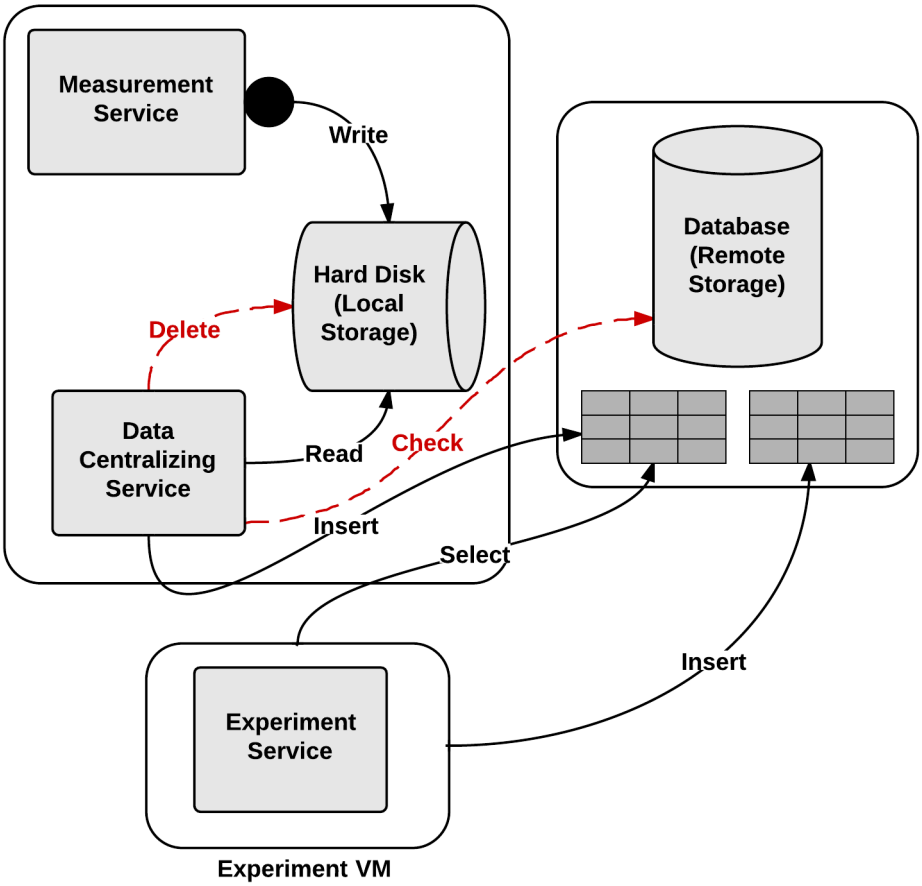



Figure 3.3: Measurement Environment.

of the scripting language such as *Perl* or *python* and the dominant functionality of that would be split into two parts:

- Measuring: The first task of *measurement service* is to obtaining the Internet path over IPv4 and IPv6 from all local ISPs to the all other ISPs of other sites. The *measurement service* have to capture some types of data which can be use for future analysis.
- Local storing: The second task of *measurement service* is to store the collected path data in the format of files on the hard disk which is separated by the version of Internet protocol and source-destination pairs.

2. **Data centralizing service**:
The *data centralizing service* is a script that transfer the collected data which have been stored locally in each site of NorNet testbed to the remote storage which is as the format of database. The *data centralizing service* functionality could be break up into three parts:

   - Checking: The first step of *data centralizing service* is to checking whether the remote storage is reachable or not.

- Inserting: If the remote storage is responsive and reachable, the second step is to reading the locally stored data file one by one and inserting to the remotely database.
- Deleting: If the process of inserting data finished successfully then the local files are needed to be remove.

3. **Experiments service**:
   The *experiment service* is a script which is be able to select the selectable data from database and run the experiments on the data and results of the experiments could be insert in the separate table in database for the future analysis.

# Chapter 4

# Implementation

In this chapter the prototype architecture would be turn into the actual result. The Internet path measurement, data collection and experiments deployment which are based on the design from the previous chapter will be propose. First the Internet path measurement implementation and the functionality of some important parts of scripts, and also some raw data will be illustrated. Furthermore the process of data collection and substantial implementation of database structure and finally the process of experiments executions will be shown. In this way, the research objective which is the observation of Internet paths and how they are in the stability perspective and also comparison analysis in the set of IPv4 and IPv6 Internet paths will be overcame.

In this fashion this chapter would be split into two following sections:

## 4.1 Measurement Implementation

Here, ***Python*** as scripting language, ***PostgreSql*** as a database server, and ***R studio*** as statistical computing and graphics have been selected. Thus, *"**NorNet-Trace**"* is the donated name to the measurement script which collects the Internet paths over IPv4 and IPv6, *"**NorNet-Trace-Import**"* is the donated name for script that do the process of importing data into the database, and ***trace-configuration*** is the donated name to the configuration file of the whole measurement. All scripts can be find in Appendix 8.1.

The implementation would be split into following sections which each describes the implementation of a service as well as the final results:

### 4.1.1 *NorNet-Trace* service

The implementation of this project starts with *NorNet-Trace* service and Internet path measurement would be done by this service. Figure 4.1 illustrates the flowchart of this measurement.

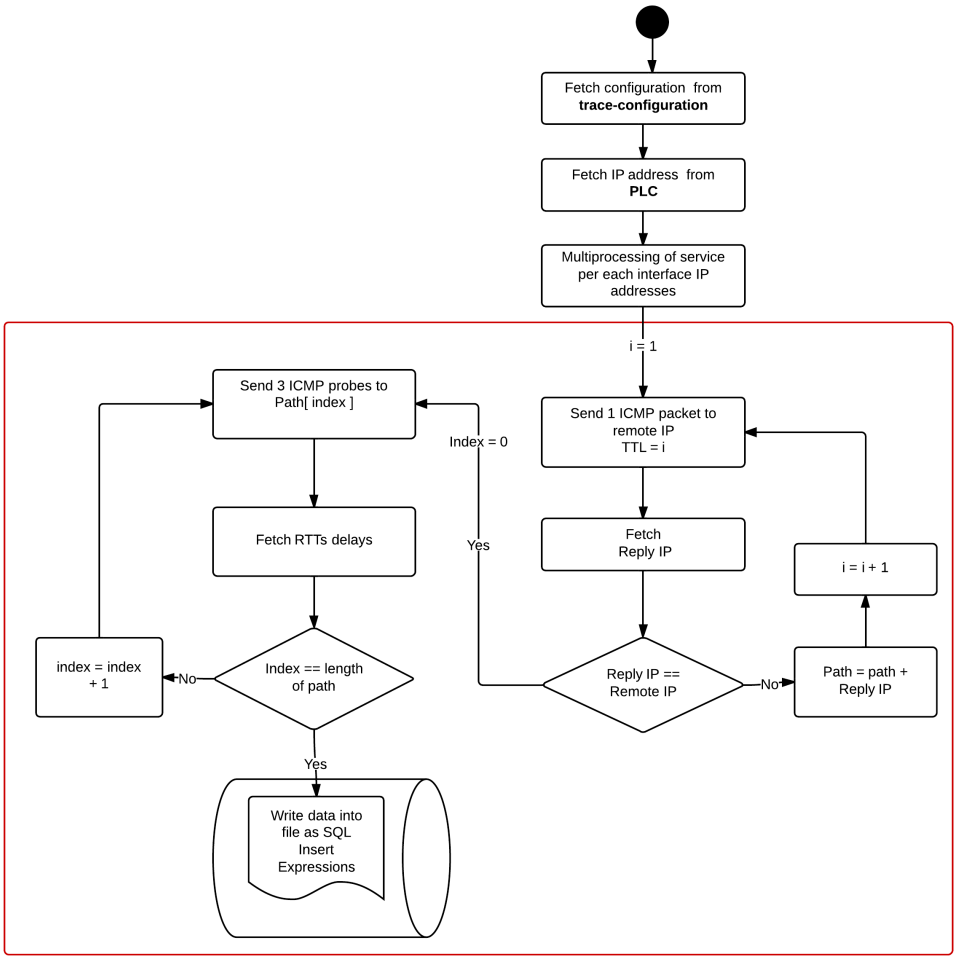



Figure 4.1: NorNet-Trace service flowchart.

The detail explanation of the *NorNet-Trace* service measurement is as follow:

#### 4.1.1.1 Fetching configuration from *trace-configuration*

*Trace-configuration* is a text file that contains all configuration information of this project such as *names of database*, *database hostname or IP address*, *port*, *username* , *password*, and *name of related tables in the database* that make it possible for each site to connect to the remote database machine. In addition *result directory path* which the collected data files will be places locally in each sites is provided by the *trace-configuration* file.

This configuration file improves the robustness of this service and would be ease any future modification or changes in services or database configurations. In the condition of configuration changes it is just necessary to modify the *trace-configuration* and other scripts would be leave unchanged.

#### 4.1.1.2 Fetching all sites IP addresses from *PLC*

NorNet Core testbed sites need to connect to the central PLC in order to get all configurations such as the configuration of interfaces as has been noted in background section 2.2.5. Here also in order to measure the Internet path from each site to the other sites the IP addresses of other sites are needed, Here the measurement script need to connect the central PLC and fetching the full list of all sites IP address.

The obtained list from PLC consist of the all *Site indexes*, and related *provider indexes* that are connected to each sites, and also *IPv4 address* of each service provider, and *IPv6 address*, if has been provided from the service provider. In this position the locals and all remote IP address are available for the measurement script.

The connecting procedure to the central PLC and fetching IP addresses is performing once each hour. Hence if there was any update in the site IP addresses and connections to the Internet Service Providers the sites IP addresses list will be updated.

#### 4.1.1.3 Multiprocessing of service per each interface IP addresses

As has been described in the previous section that the complexity of the Internet path measurement in each site is equal to capturing Internet path from all local interfaces per each IPv4 and IPv6 addresses, to all interfaces IP address of remote sites 3.1. Hence, in order to accelerate the measurement procedure it is a wise decision to run the measurement simultaneously per each Internet protocol versions of local interfaces. This process have been done by ***forking*** each Internet protocol versions of all local interfaces.

For instance if local site connected to three ISPs and two of ISPs provide both IPv4 and IPv6 connections and one of them provide connection just over IPv4, therefore the service forking each IP addresses and create five sub processes that measure Internet path to other sites concurrently and the measurement are the same for each sub process.

#### 4.1.1.4 Measuring the Internet path

The next step after forking the interfaces per each IP addresses and having the source and destination IP addresses, is to sending a probe from local source IP address to remote IP address. The *NorNet-Trace* service uses *Ping* command in order to capture the Internet path while the detail specification and the way that *ping* utility works has been explained in 2.3.2.

This measurement is done distinctly for each source-destination IP address pairs. Thus the following explanations illustrates the measurements

for a single source-destination pair which is the same for the other source-destination pairs.

The Measurement starts with sending an *ICMP* probe with *TTL (Time-To-Live)* value one (TTL = 1) from source to destination. Setting the *TTL* value to one cause that the first router in the path return the *ICMP Time Exceeded reply* packet with IP address of itself. Hence the IP address of first intermediate router is captured. The process of sending *ICMP* request probe to destination IP and capturing the IP address of reply probe is continued by increasing the *TTL* value until reaching the destination IP as has been shown the Figure 4.2. By sending ICMP probe with *TTl = 1*, The IP address of router *1*, *TTl = 2* the IP address of router *2* and so on will be captured.

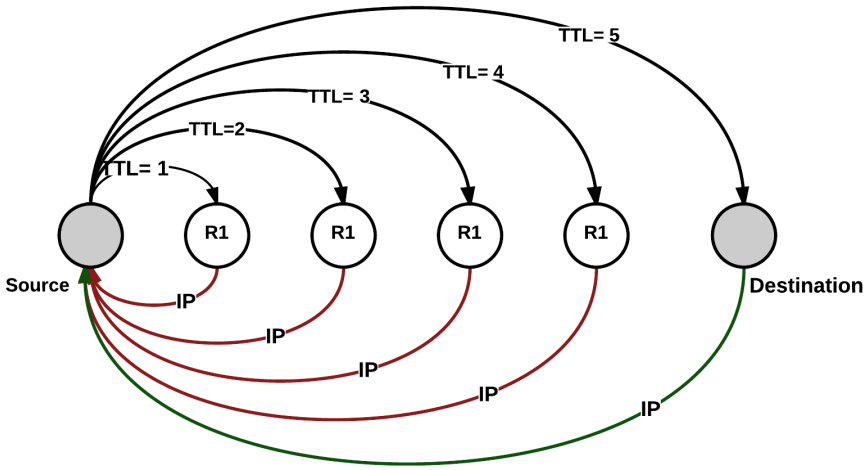



Figure 4.2: Internet Path measuring by *ping*.

The *ping* command which has been shown below, sending just one packet *(**-c -1**)*, and waits just for two seconds *(**-w 2**)*. Setting the *ping* command to waits just two second prevent the ping waits for a long time *(default ping timeout is two RTTs)* in the condition that one of the intermediate routers has been configured to answer to a limited or non *ICMP* packets, and Leeds into the service takes long or fails in finding the Internet path.

*Ping* command in *NorNet-Trace* service

```
1  ping -n -w 2 -c 1 -t <Variable>  -I  <Source IP Address><Destination IP Address>
```

As can be seen from the *ping* command above the *Numeric output only (**-n**)* option causes that the ping command does not looking for host name of routers and just shows the IP address of them. This option speeds up the measurement time and prevents wasting some times in order the ping to lookup the symbolic name of host address.

As has been explained in the background section that some routers in some networks are *anonymous routers* and has been configured to answer

to a limited amount of *ICMP* packet or not respond to any *ICMP* packet at all, as the reason of network security or performance 2.3.3.3, the IP address of some intermediate routers is undetectable. In such a condition that the intermediate routers does not responsive for any reason the *NorNet-Trace service* has been configured to put a star *('*')* instead of that. Therefore any star in the path shows the existence of a router that could not respond to the *ICMP* packet.

This process is done in a loop while in each iteration the *TTL* value is increased and continued until *TTL* value *35*. The selection of *35* as *TTL* value is due to the long distance between china site and Norway site. There could be more than **30** (ping default *TTL* value) intermediate routers between sites of Norway and China. In the condition that the destination site was not reached by *TTL* value of *35*, *route problem* or *destination unreachable* will be deducted.

The IP address of each intermediate routers between source node to destination node is captured in this way and the Internet path or Internet route between specific source-destination pair is the exact series of obtained IP addresses. The number intermediate hops is now elicited for this Internet path.

#### 4.1.1.5 Measuring RTTs delays

After having the complete intermediate routers IP addresses between a source-destination pairs or in another work the Internet route, the RTT *(Round-Trip-Time* delays which shows the length of time that a packet takes to be sent from a source and reached the destination, plus the length of time for the acknowledgement that the packet received. It means the time between the *ICMP request* probe leaves the source node until reaches the destination node, plus the time that the *ICMP reply* probe leaves the destination node until reaches the source node. The process of measuring and capturing the *RTT* values will be divided into two parts:

- **Hop-by-hop delays**:
  The *RTT* delays from the source node to each router in the path separately, toward the destination node can be important. In this fashion the preference of each single router can be elicited from the *RTT* delays.

- **End-to-End delays**:
  The *RTT* delays between the source and destination nodes shows the route performance and the time length that a packet takes in order to reaches the destination by using that path.

*Ping* command shows a statistical summary as a result after completing the probes and illustrates *minimum, average, maximum* and *standard deviation*

values between the number of ping probes RTTs that has been sent. Hence we use the *ping* utility to obtain the RTT values which starts by sending three *ICMP* probes to the sequence of IP addresses of captured path, from the first IP address til the last IP address.

This process is done in a loop and in each iteration the *ping* command is run from source IP address to the specific intermediate routers IP address in the path and the RTTs is captured in the format of *minimum, average, maximum* and *standard deviation* and each these values is added in separate text variable. It means the specific index in *minimum, average maximum* and *standard deviation* variables shows the exact values of RTTs to the according routers IP address index in the captured path.

As we noted before , it is more likely that we see star *('*')* in some paths as the reason of anonymous routers in some networks. In this condition fetching the RTTs delays is impossible and we also put star *('*')* instead of RTT values for correlated IP address.

Finally the RTT values from source IP address to the destination IP address, or in another word *end-to-end* delays is captured which we shows it as *total minimum, total average, total maximum* and *total standard deviation*.

#### 4.1.1.6 Storing on Hard disk as text file

The obtained data such as Internet path and RTT delays between a source-destination pairs need to be stored for the further processing. As noted before in the previous section the data needs to be stored locally on the hard disk before centralization process and transferring them to the database. Each source-destination measured data is stored as a the text file distinctly.

In order to prevent any interference and data overwrite, it is needed to use a format for naming the files which make the name of them unique. In this way we use following format for measurements data files naming:

Local data files naming format.

```
route<Version>-<Remote site index>-<Local Provider Index>-<Remote provider index>
-<date and time>
```

In addition to the unique data file naming, this format provide an indication of source-destination, Internet protocol version and time of measurement by:

- **Version**: The indicator of IP version that the data had been captured over. I could be **4** for IPv4 or **6** for IPv6.
- **Remote site index**: The index of destination site.
- **Local provider index**: The index of source Internet provider on the local site.

- **Remote provider index**: The index of destination Internet provider on the remote site.
- **Date & Time**: The exact date and time that the measurement ran and data collected.

The combination of these data make the name of file unique. The process of storing data files locally consists of two parts:

- Types of data that should be collected and stored in text files locally for the future analysis are illustrated in the Table 4.1.

| Data Type | Description |
|---|---|
| Date | Date and time of data collection in the format of <YYYY-MM-DD hh:mm:ss.fffff> |
| Version | Internet protocol version [ 4 , 6 ] |
| FromSI | Source Site Index (Local Site Index) |
| FromPI | Source Provider Index (Local Provider Index) |
| FromIP | Source IP address |
| ToSI | Destination Site Index (Remote Site Index) |
| ToPI | Destination Provider Index (Remote Provider Index) |
| ToIP | Destination IP address |
| Path | Sequence of intermediate IP addresses between source and destination |
| PathID | Path hash with MD5 hashing algorithm |
| HopNumber | The number of intermediate routers in the path |
| PingNumber | Number of sent ICMP probes by ping command |
| Min | Sequence of RTT's minimum to the sequence of intermediate IPs |
| Avg | Sequence of RTT's average to the sequence of intermediate IPs |
| Max | Sequence of RTT's maximum to the sequence of intermediate IPs |
| Std | Sequence of RTT's standard deviation to the sequence of intermediate IPs |
| Scheme | The type of probes (ICMP = 1) |
| TotalMin | Minimum of RTTs from source to destination |
| TotalAvg | Average of RTTs from source to destination |
| TotalMax | Maximum of RTTs from source to destination |
| TotalStd | Standard deviation of RTTs from source to destination |

Table 4.1: Types of data that need to be stored.

- Format of storing :
  As noted before the stored data files is needed to transfer to the database and centralized from time to time. Therefore a method of storing data in the files is needed which makes the the centralization process easier and accelerate this process. In this fashion the decision has been made to store the data in the text file in the format of **Insert SQL expressions** which has been shown below:

Insert SQL expressions format in the data files.

```
INSERT INTO <TABLE>
(<Table Field 1>,<Table Field 2>,<Table Field 3>,..., <Table Field n>)
VALUES (<Value 1>,<Value 2>,<Value 3>,...,<Value n>)
```

The $Table \quad Field_n$ are the same data types illustrated in the Table 4.1, and the $Value_n$ are their correlated values.

The *Measuring the Internet path (4.1.1.4) and Measuring RTTs delays (M2)* sections are repeated for each local IP addresses to other the remote sites IP addresses (IPv4 & IPv6) while all the measuring procedure is done for each local IP addresses simultaneously. This loop has been shown by the red rectangle in figure 4.1. When the measurement is finished for a round from all local IP addresses (IPv4 & IPv6), it will be sleep for approximately 2 minutes (between 90 seconds to 150 seconds) and then starts measuring.

#### 4.1.1.7 Measurement repetition

The Internet path measurement of the *NorNet-Trace* service at this time with current sites and current source-destination pairs taking about **10** minutes to finish one iteration. While each run of the service has been finished the measurement sleeps for around **2** minutes, hence the Internet path between each source-destination pairs will be measured and data will be provided based on the table 4.2.

| Time | Iteration of NorNet-Trace Service |
|---|---|
| 1 hour | 5 |
| 1 day | 292 |

Table 4.2: *NorNet-Trace* service iteration in time period.

### 4.1.2 *NorNet-Trace-Import* service

As has been noted in the previous section that the stored data file in each sites needs to be centralized in database due to the fact that sites have access just to routing file from local interfaces to the other sites. In addition the process of centralizing should be done with a time interval which could be effective in providing up to date routing data, and not increases the network and I/O load.

The operation of transferring Internet paths data to the central database is done by *NorNet-Trace-Import* service which is performed in each site distinctly. The functionality of this service has been summarized in figure 4.3 and the detail description has been divided into several parts and provided as following .

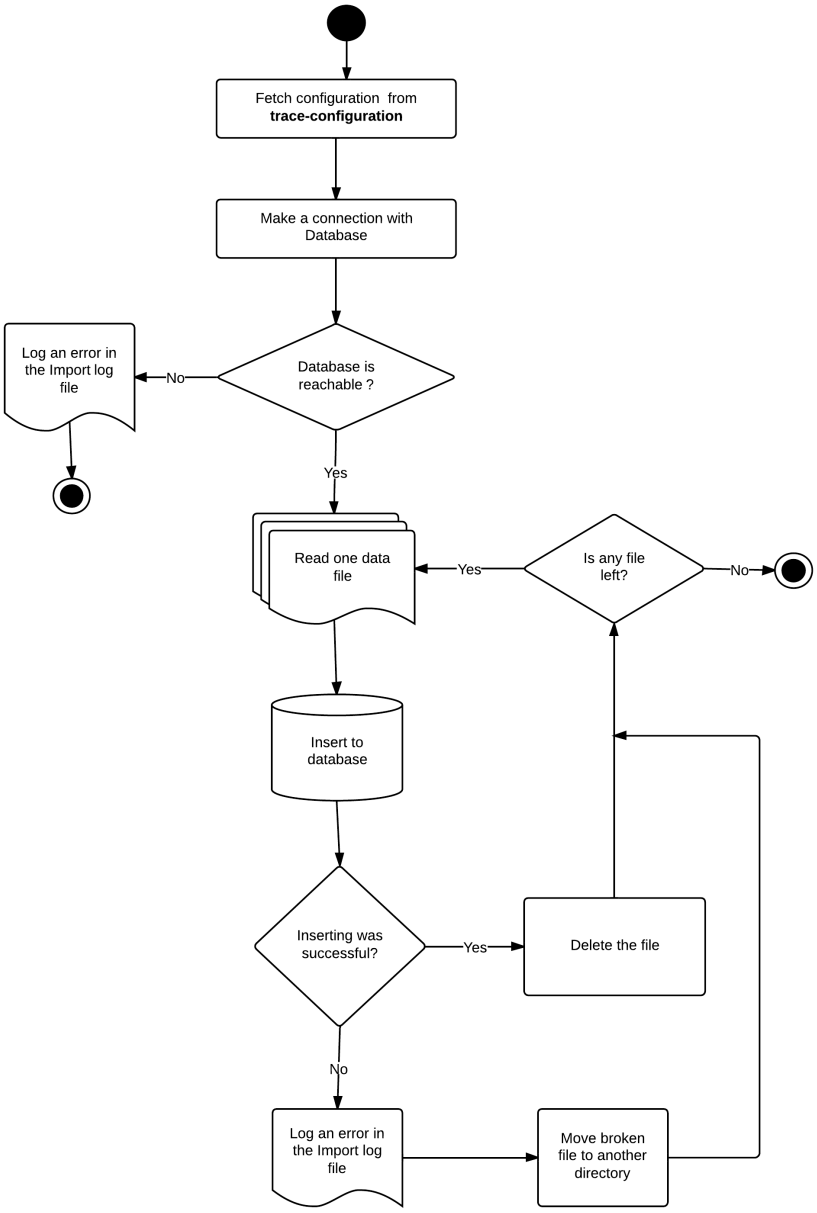



Figure 4.3: NorNet-Trace-Import service flowchart.

#### 4.1.2.1 Fetching configuration from *trace-configuration*

The first step to accomplish the process of transferring routing data into the database is getting required preference from *trace-configuration* file. These configuration consists of **database machine IP address or hostname, port number, database name, user name, password, table name** and *result directory path* which contains the routing files.

#### 4.1.2.2 Make a connection with database

A connection to the database is needed which can be done one time in each iteration of import service performs. This connection is done by following command in python by using ***psycopg2*** python library, which use *database host name, database name, port number, user name and password* for accessing the database.

Connection to the *database*.

```
Conn = psycopg2.connect(dbname=<database name>,
                  user=<user name>, host=<host name>, port=<port number>,
                  password=<password>)
```

- ***Database* is not reachable**:
  If the database could not be reachable for any reason, the service will be stop until reaching next time interval and log the error in the ***log file***.
- **Connection is established**:
  After the setting up the connection to the database server the Importing data process into the database can be start.

#### 4.1.2.3 Inserting data into the *database*

This part of *NorNet-Trace-Import* service is perform for all routing data files one by one in a loop as has been showed in the Algorithm 1.

**Algorithm 1** Import data into the database

```
for all (file) in result directory do
  line = read(file)
  if not (execute (line)) then
    Log(error)
    Move (file) to (error-files) directory.
  end if
end for
```

In each data files in the *result directory* there is just one line for a single Internet path that has been measured by *NorNet-Trace* service and stored locally in *result directory* of each site. On top of that, as has been explained before the routing data has been stored in the format of *SQL INSERT* expression. In this way no additional process is needed for inserting the routing data to the database except reading the routing files in executing exact the line which has been read from the routing file.

As we can see from ***for loop*** in Algorithm 1, if the routing data can Insert into the database, it means the *SQL INSERT* expression and the captured data is in the right form. After inserting the data successfully into the database, there is no need to keep the files in the *result directory*, hence the

local files should be deleted after each successful database import.

In the condition that the insert action could not be performed means there are some malformed data in the file which are not match with the fields type of database table, therefore a *log* message is needed to be inserted into the *log file*, plus a moving action which moves the malformed file to another directory (*error-file* directory) in order to prevent future processes on the malformed file in the next iteration of service runs.

The *NorNet-Trace-Import* service is running as a cron job, once per hour and inserting all measured routing data into the database.

The *"**route**"* is the donated name to the table in the database which contains the Internet paths derived from the measurement.

## 4.2 Internet Path change extraction

After obtaining the Internet paths and stored centrally in the database which includes the Internet paths from each sites to the all other site in the size of connection mesh which has been noted in section 3.1, it is necessary to extract the Internet path changes from the original Internet path data that has been inserted in the *'route'* table. In order to show the behaviour of Internet routes in stability perspective it is need extract the route changes as follow:

### 4.2.1 Internet Path changes classification

In the connection between a source-destination pair, any path changes from the first record of measurement til the last could be recognized by the *PathID* field in the *route* table of the database which has been explained in Table 4.1. The similarity of *PathID* between two records in the *route* table for a single source-destination pair shows the equality of the Internet route of those two iteration of measurement executions. Thus difference in *PathID* field shows inequality of Internet route.

The first diagnosis step of the Internet path changes is two extracting the changes records from the original data table (*route* table) and classifying the changes based on cause of changes. The extraction of changes can be done by comparing the *PathID* field. When there is a difference in *PathID* between two consecutive records of a source-destination pair a flag will be raised that could be split into three different changes as follow:

#### 4.2.1.1 Hop count changes

One of the changes that could be seen in the selected data is the changes in the number of intermediate routers between a source-destination pairs and we name it ***Hop count change***. The *Hop count changes* would be appeared by increase or decrease in the number of intermediate router from source node to the destination node.

In the *route* table the number of intermediate routers between a source-destination pair is stored as *HopNumber* field in each iteration of measurement runs.

#### 4.2.1.2 IP changes

Another change can be the changes in the IP addresses of the intermediate routers. The number of intermediate hops can be the same in two single runs of measurement but it is possible that one of the intermediate router change its place with another router, Therefore the IP address of one specific index in the *Path* field will be change to another IP address in the next iteration. The figure 4.4 illustrate the Constancy of *HopNumber* field but with different IP addresses of different intermediate routers in the Internet path of two iteration of measurement executions.

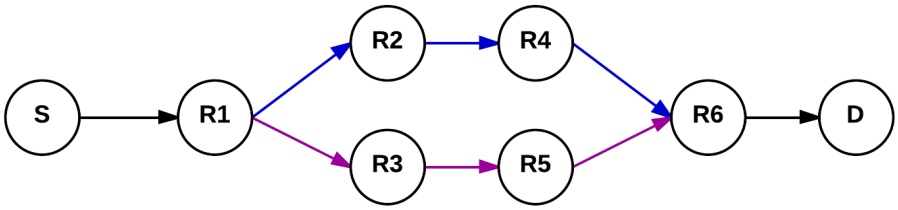



Figure 4.4: IP address change example.

In this condition it is necessary to traversing the *Path* field of the two selected records from *route* table which is the sequence of IP address of the Internet path between a source-destination and comparing each index of *path* field of first record to the second record. As has been shown in Figure 4.4 the IP address of R2 has been changed to R3 and the IP address of R4 has been changed to R5.

#### 4.2.1.3 Star changes

Another type of change is when the *Hop Number* has no changes and the IP address in the specific index of *Path* field has been changed to the star ('*') as has been noted before in section 4.1.1.4. These kind of changes would be added in *star changes* class that star ('*') could be the same or different

router which have been configured to answer to a limit amount of *ICMP* probes, or not to be responsive at all.

It is possible that in some change records we observe both *IP change* and *Star change* which these changes will be included in both *IP change* and *Star change* classes. In order to make future analysis more convenient we coded the classes of changes based on *Hop changes*, *IP changes*, and *Star changes* which illustrate in Table 4.3.

| **Comparekey** | **Description** |
|---|---|
| 1 | Hop changes |
| 2 | IP changes |
| 3 | Star changes |
| 4 | IP + Star changes |

Table 4.3: Codes of change classes.

The Internet path changes between all source-destination pairs are extracted from *route* table and will inserted in the new table with donated name ***compare***. Based on the aims of this research which is a comparative study, *compare* table would be quite useful for future analysis because it is consist of the exact Internet path changes which we need. Hence it is necessary to have enough information in this table that can be used in this comparative study.

In addition to the *CompareKey* field which consists of codes of change classes, the index of IP or Star change in the Internet path which shows the exact index of intermediate hop that had changes, and also the record number of *route* table which had changes are extracted. Table 4.4 illustrates the fields of *compare* table with description.

| **Data Type** | **Description** |
|---|---|
| Date | Date and time of changes in the format of <YYYY-MM-DD hh:mm:ss.fffff> |
| Version | Internet protocol version [ 4 , 6 ] |
| FromSI | Source Site Index (Local Site Index) |
| FromPI | Source Provider Index (Local Provider Index) |
| ToSI | Destination Site Index (Remote Site Index) |
| ToPI | Destination Provider Index (Remote Provider Index) |
| CompareKey | The code of change class |
| IPIndex | Index in the path that had change in IP address |
| Star ex | Index in the path that had change in Star address |
| ChangeIndex | Index of record in the *route* table that had change |

Table 4.4: Field of *compare* table.

## 4.3 Analysing the results

Based on the designs of the previous chapter, some data processing are needed to be performed in order to study how Internet path changes over IPv4 differs from IPv6 changes.The Internet path changes which are stored in ***compare*** table is the base of all data processing procedures which will be explained later. Here there are some points which should be considered.

1. As has been noted before in NorNet Core testbed, IPv6 is not available from some Internet service providers in some sites the table 4.5 illustrates the number of source-destination pairs connections over IPv4 and IPv6 pairs.

| Internet Protocol | Source-Destination pairs |
|---|---|
| Total IPv4 connections | 722 |
| Total IPv6 connections | 292 |
| IPv4 connections without correspondence IPv6 connections | 430 |

Table 4.5: The number of source-destination pairs connections over IPv4 and IPv6 in NorNet Core testbed.

   Therefore, the analysis is done separately for the two types of source-destination pairs:

   - Measurement of Internet routes just for sites that both IPv4 and IPv6 are available:
     In this way we just exclude the IPv4 source-destination pairs that there is no IPv6 correspondence for that which can provide equal amount of source-destination pair for providing fairly compression.

   - Measurement of Internet routes for All sites Over IPv4 and IPv6:
     Here we include all IPv4 source-destination pairs in order to broaden the experiment base.

   All data processing from this time on will be proposed in two formats, one for *all sites* regardless of version of Internet protocol, and second for just sites that have IPv6 connections available.

2. Excluding Star changes from all data processing procedures:
   The star ('*') observation in the Internet paths are likely due to the configuration of some intermediate routers to answer to a limited amount of *ICMP* probes or not to be responsive at all as has been explained before in the section 4.1.1.4. Therefore the *start change* class

which has been captured and stored in *compare* table is more likely due to the *ICMP limitation* and is not the real Internet path change. Although this change type is also could be useful and informative but extracting the real path changes from this class need more investigation which could not be incorporated in this time. Hence in the following data processing procedures the *star change* class has been excluded, and they have been measured for *Hop change*, *IP change* and *Hop + IP change* classes.

### 4.3.1 Distribution of path length

The Internet path length means the number of intermediate router between a source-destination pairs or in another word *HopNumber* field in the *compare* table. It is possible to illustrate the length of Internet paths over IPv4 and IPv6 by getting the path length between each source-destination pairs and observing how it is distributed. The path length can be fetched from *compare* table, distinctly for each source-destination over IPv4 and IPv6 by following SQL select expressions:

Path length selection from *compare* table.

```
SELECT distinct(HopNumber) FROM (compare) WHERE FromSI =<fromsiteindex>
         AND FromPI =<fromproviderindex> AND ToSI = <tositeindex>
          AND ToPI = <toproviderindex> AND Version = <version>
```

The output data of this selection is the exact number of hops in the path from a source to a destination which could be more than one value if there were changes in the number of intermediate routers between that source-destination pair. By getting these value separately for each source-destination combinations over IPv4 and IPv6, will resulted in a comparative study how the Internet path length differs over IPv4 and IPv6.

### 4.3.2 Path change average per day

We calculate how many path changes each source-destination pair experienced per day in average. This average could be calculated by dividing *total path changes* by the *total measurement days* which could be fetched from original *route* table. This type of data processing should be done for all kind of change classes (*hop changes*, *IP changes*, *Hop + IP changes*) that have been shown in Table 4.3. The process of obtaining the average of path changes per day for all path change types follows Algorithm 2.

**Algorithm 2** Path change average per day.

```
for all Path change classes do
  for all Versions do
    average = Total change / Total days
  end for
end for
```

The average value would be calculated for all source-destination pairs separately based on *Hop changes*, *IP changes* and *HOP + IP changes*, over IPv4 and IPv6. The *total measurement days* would be fetched from *route* table of the database by subtracting the *End date* from the *Start date* of measurement and the *total change* value would be extracted easily from *compare* table, while Figure 4.5 illustrates this process.

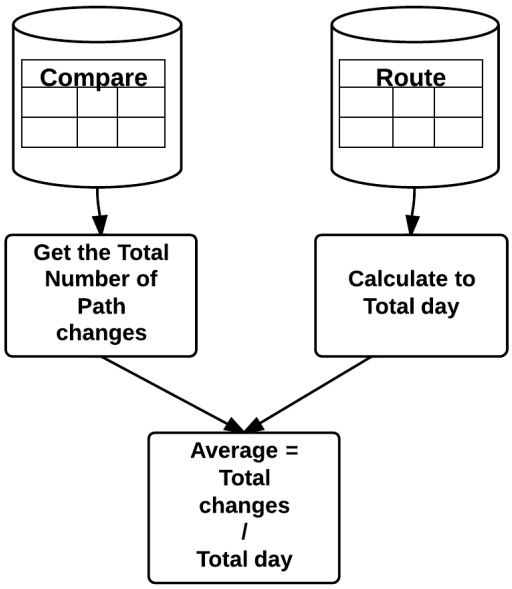



Figure 4.5: Path changes average per day.

### 4.3.3 Path change distribution

Next experimental data processing which is the path change distribution will show how Internet path changes are distributed in change days. This would be obviated by fetching the exact number of days that experienced path changes in a single source-destination pair from *compare* table of database, plus the exact path changes for that specific source-destination.

This data processing would be done for all source-destination pairs distinctly and all different classes of changes such as *Hop changes*, *IP changes* and *Hop + IP changes*, over IPv4 and IPv6.

This step would be accomplished by getting the *total measurement days*

from *route* table like previous section and extracting the *total change days* and *total changes* from *compare* table which Figure 4.5 illustrating. In addition the distribution of *Internet path changes* in *change days* will follow Algorithm 3 which would be perform for all source-destination pairs.

**Algorithm 3** Path change distribution in change days.

```
for all Path change classes do
  for all Versions do
    Fraction of change days= Change days / Total days
    Fraction of changes = Total changes / Change days
  end for
end for
```

The purpose of extracting this metric is to investigating whether path changes happen regularly every day, or if they come as large bursts in just a few days.

### 4.3.4 Path changes grouping

A data processing would be partitioning the Internet path changes of a source-destination pairs into events. The events would be create by defining a time threshold (in hour) and adding the path changes into an event which there are a *time difference* between two consecutive events is equal or less than the threshold time. It means the path changes between a source-destination pair would be split into one or more events while there is a time interval more than threshold time between each event.

An event in this context is a group of path changes that take place in sequence such that there is never a long period of silence between two consecutive changes. The purpose of this grouping is to describe the duration and size of a path instability. In particular, we expect that paths with load balancing will have very long lasting events.

Algorithm 4 explain this process more clearly.

**Algorithm 4** Grouping the path changes into event.

```
for all Path change classes do
  for all Versions do
    if Time-difference < threshold-time then
      Add to the Current Event
    else
      Create New Event
    end if
  end for
end for
```

In this fashion each event contains a number of path changes as *event*

*size*. Beside the *event size* the *event duration* which propose how long an event has been taken.

In overall the Internet path changes will be split into the big size events which are the long lasting events with many changes and small size of events which are short lasting events with small amount of changes. Afterwards these events would be grouping into source-destination pairs with small amount of big size events, and source-destination pairs with many short size events as illustrated in Figure 4.6.

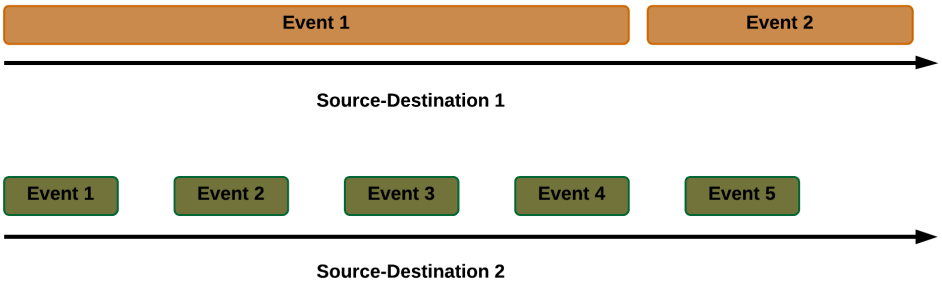



Figure 4.6: Events size and duration in two different source-destination pairs .

This grouping action of events would be provided a distinction of *real path changes* to the changes which caused by *load balancer*.

# Chapter 5

# Result & Analysis

In this the chapter the actual results of the data processing on raw data in database will be turned into to plots while the description of each plot besides their analysis will be provided here. This chapter will be divided into the several section based on the design of different data processing which have been proposed in previous chapter.

## 5.1 Path Length Distribution

The *path length distribution* provides an informative knowledge about how many intermediate routers are existed in average between source-destination pairs connection and display how source-destination pairs are distributed in the different path length. It provides a comparative study of Internet path length over IPv4 versus Internet path length over IPv6, and how they differ in the path length perspective. Figure 5.1 illustrates the distribution of Internet path length over IPv4 and IPv6.

As we can see from the plot 5.1, the *blue* line shows the distribution of Internet path length of source-destination pairs over IPv4 that have correspondence IPv6 connection, and the *red* line displays the Internet path length distribution over IPv6. The *blue* line and *red* line show the connections between the same source-destination pairs which connectivity over both IPv4 and IPv6 are available for them. In additional the *green* line illustrates the Internet path length distribution between all IPv4 connections regardless of Ipv6 availability.

The *X-axis* of the plots displays the number of path length or in another word the number of intermediate router between the source-destination pairs, while *Y-axis* shows the probability of distribution of path length of each source-destination pairs on each *x-axis* value.

The first plot in figure 5.1 is a cumulative probability plot which proposes a cumulative distribution function (***CDF***) of the Internet path length over IPv4 and IPv6. As we can see from the *CDF* plot which illustrated the collected Internet path length on the x-axis against the fraction of source-

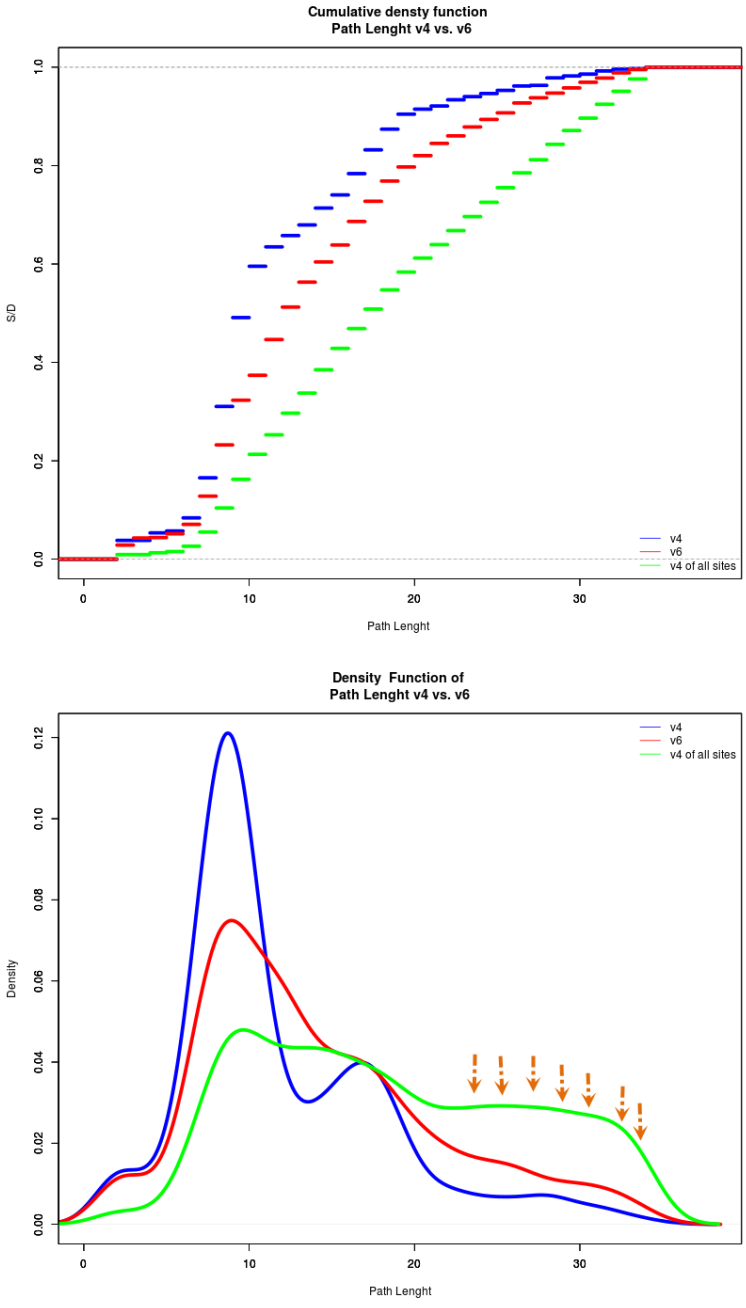



Figure 5.1: Path Length distribution over IPv4 and IPv6..

destination pairs which have Internet path length equal or less than the value on the *x-axis*.

As the Internet path length are discrete values the cumulative probability function follows formula 5.1:

$$F(x) = P(X \leq x) = \sum_{x_i \leq x} P(X = x_i) = \sum_{x_i \leq x} P(x_i) \tag{5.1}$$

$$P(a \leq x \leq b) = P(b) - P(a) = \sum_{x_i \leq b} P(X = x_i) - \sum_{x_i \leq a} P(X = x_i)$$

The CDF plot in figure 5.1 shows that *60%* of the source-destination pairs have less than 10 intermediate routers over IPv4 connections while just *40%* of connections over IPv6 have less than 10 hops between source to the destination.

It is also observable that around *92%* of connections over IPv4 have less than 20 path length which means *32%* of them are distributed between 10 and 20, while *80%* source-destination connections over IPv6 have less than 20 path length which means *40%* of them are distributed between 10 and 20 path length.

It is also noticeable that *20%* of source-destination pairs have more than 20 path length over IPv6 while just *8%* of IPv4 connections have more than 20 intermediate routers in their connections.

The second plot in figure 5.1 is the probability mass function (***PMF***) or discrete density function, which shows the probability of source-destination pairs having an exact Internet path length on the *x-axis*. As we have the Internet path length as discrete values, the PMF following formula 5.2:

$$f(x) = P(X = x) \quad for \quad all \quad x \in S \tag{5.2}$$

$$f(x) \geq 0$$
$$\sum_{x \in S} f(x) = 1$$

It is observable from the PMF plot that the probability of that the source-destination pairs connection over IPv4 have path length equal to 10 is *11.5%* while it is *7.5%* for source-destination connections over IPv6. The plot shows also the probability of the IPv4 connections path length equal to 22 intermediate routers is around *1%* while it is almost *3%* for IPv6.

In this fashion if we want to see the probability of source-destination pairs connection less than or equal a single path length from the PMF plot, it is necessary to calculate the sum of each single probabilities which are less than and equal the path length on the x-axis, while provide the same result of the CDF plot.

The *green* curve from the both CDF and PMF plot in figure 5.1 illustrates the probability distribution of all source-destination pairs connection over IPv4 regardless of correspondence connectivity over IPv6. As has been noted before that there is lack of connectivity over IPv6 from some providers in several NorNet core sites, while some long distance sites such as *China* and *Versatel* Internet provider in *Germany* site are included in those connections. The sites and their providers and also the total number of source-destination pairs over IPv4 and IPv6 can be seen from table 2.1 and table 4.5 respectively.

The additional connections over IPv4 led into the source-destination pairs connections over IPv4 (*green* line) distributed in all different path length. In addition we can see from the selected area with flashes from the *green* line distribution in PMF plot, more source-destinations pairs are distributed between 22 and 34 path length. Whereas the connections from all NorNet Core sites to the *china* site (both *CERNET* and *China Unicom* providers) and also reverses connections from *china* site to other sites are mostly distributed in that area. Hence caused from the long distance of Europe sites (*Norway, Sweden, Germany*) and the *china* site. It is noticeable that the lack of IPv6 connection in several sites especially long distance sites provide this difference between IPv6 (*red* curve) and all site connection over IPv4(*green* curve).

Main observation from the *Internet path length distribution* data processing which has been shown in figure 5.1 are:

- Most of Internet paths are distributed between **5** and **20** hops.
- IPv6 paths are generally longer than the IPv4 paths. In equal connections over IPv4 and IPv6.
- The long paths to sites like *China* and *Germany* makes IPv4 of all sites(*green* line) paths longer.
- The reason of whether IPv6 paths are longer is unclear, while one explanations could be that the IPv6 topology is more sparse, therefore more links in the IPv6 paths which makes it longer.

## 5.2 Path Change Average per day

The average value of each source-destination pairs path changes per day illustrates that how many path changes each source-destination pairs have been experienced during the measurement time. It is also provide a comparative study about the Internet path changes over IPv4 versus IPv6 and shows which one of the Internet protocols have more path changes. In this fashion the experiments has been done based on the different classes of changes (*Hop changes, IP changes, HOP + IP changes*) as has been noted before and in addition the measurements is divided into two sections as follow:

### 5.2.1 Path change average per day of only source-destination pairs that have IPv6 available

As has been noted before IPv6 connection from some providers of some NorNet Core sites is not available, therefore in this section the *path changes average per day* results is illustrated for just source-destination pairs that have connectivity over both IPv4 and IPv6. The results out of this experiment has been proposed in *Cumulative Density Function (CDF)* while the x-axis displays the average of path changes per day against the probability of distribution of source-destination pairs on Y-axis.

As the average values on the x-axis are continuous values, therefore the *cumulative probability* follows formula 5.3:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^{x} f_x(t)dt \tag{5.3}$$

$$F(b) - F(a) = P(a \leq x \leq b) = \int_{a}^{b} f_x(t)dt$$

Figure 5.2 propose the all different measured path changes base on *Hop changes, IP changes* and *IP + Hop changes*. The *Blue* line shows the distribution of source-destination pairs connections over IPv4 and the *red* line displays the IPv6 connections distribution.

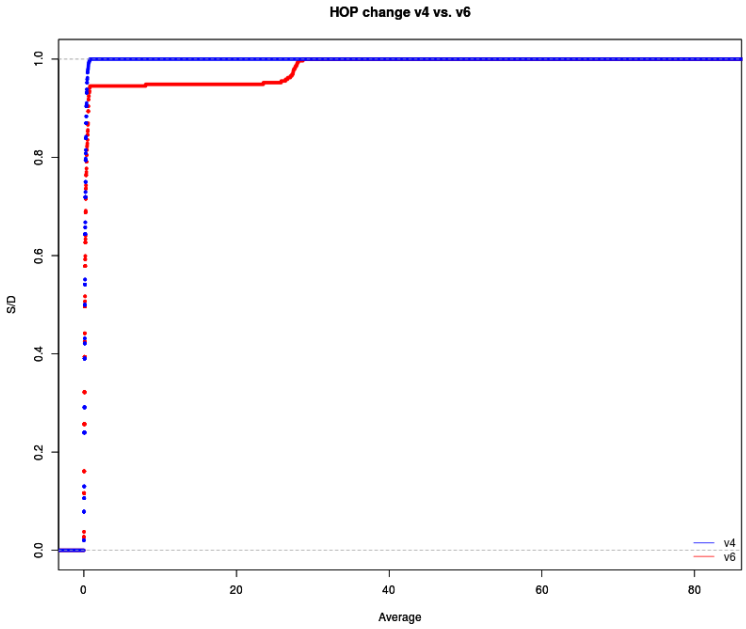
The first plot in figure 5.2 shows the probability distribution of source-destination pairs on Internet path changes based on the *Hop changes* over IPv4 and IPv6. It is observable that most of the source-destination pairs are quite stable in the *Hop change* class while approximately *93%* of IPv6 source-destination pairs have less than 1 changes per day in average, and just around *7%* of IPv6 connections have more than 30 changes per day.

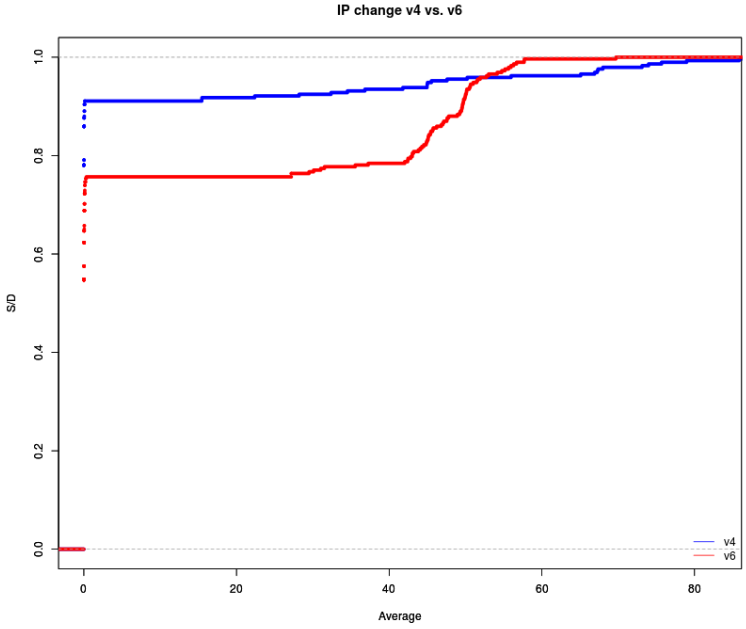
The IPv4 source-destination pairs are quite stable than IPv6 while we can see that almost all of IPv4 source-destination pairs connections have less than 1 hop changes per day in average.

The second plot in the figure 5.2 illustrates the distribution of source-destination pairs connections over IPv4 and IPv6 on the Internet path changes which are based on the *IP changes*. It is observable that the connections between all source-destination pairs experienced more IP based path changes than Hop based path changes.

As we can see from the second CDF plot almost *75%* of IPv4 connections and *55%* of IPv6 connections have zero (*0*) IP changes per day. In addition it is visible that *91%* of source-destination pair connections over IPv4, and *76%* of IPv6 connections are quite stable and have had in average less than 1 changes per day. It is also observable that *5%* of IPv4 connections have more than 50 changes per day while it is about *18%* for IPv6 connections.

The third CDF plot in figure 5.2, provides an overview of distribution of source-destination pairs on the Internet path change per day based on



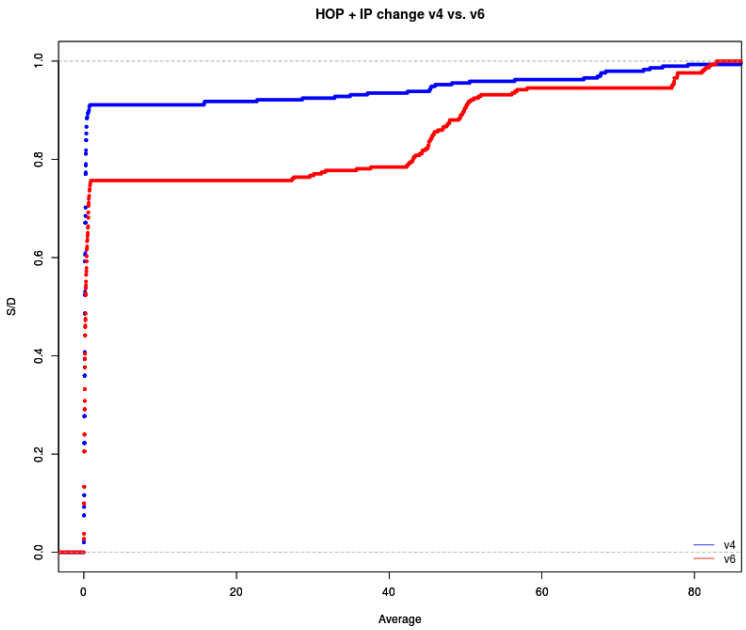




Figure 5.2: Internet path change average per day of IPV6 available sites over IPv4 and IPv6 .

both *Hop and IP changes*.It is noticeable that connections over IPv4 are more stable than connections over IPv6. For instance we can see that around *10%* of IPv6 connection have more than 80 path changes per day while it just *1% or 2%* for IPv4 connections.

**The overall observation from figure 5.2**:
It is clear that the source-destination pairs can be split into two separate groups based on the number of Internet path changes they experience. Most connections experience less than one path change per day, while others see tens of changes per day. This high frequency of changes is a clear indication that at least one router on the path is doing load balancing. With load balancing, there is a high probability that we will see a different path each time we run the measurement service.

*91%* of IPv4 source-destination pairs have less than **1** path change per day on average, while the remaining *9%* source-destination pairs have more than **15** changes per day. For IPv6, *76%* of the source-destination pairs have less than **1** change per day, while the remaining *24%* have more than **27** changes per day. From this we conclude that load balancing is more common for IPv6 than for IPv4: we see load balancing in *24%* of IPv6 paths, but only *9%* of IPv4 paths.

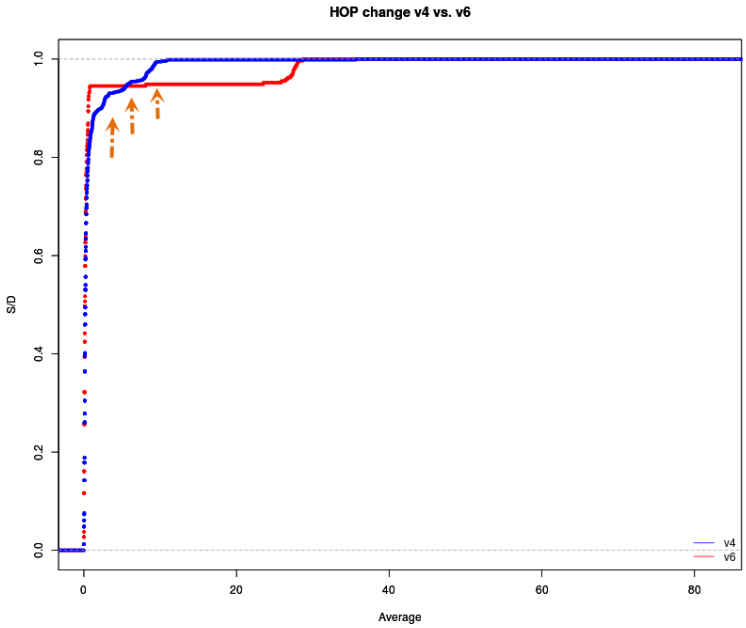
### 5.2.2 Path change average per day measured for all source-destination pairs

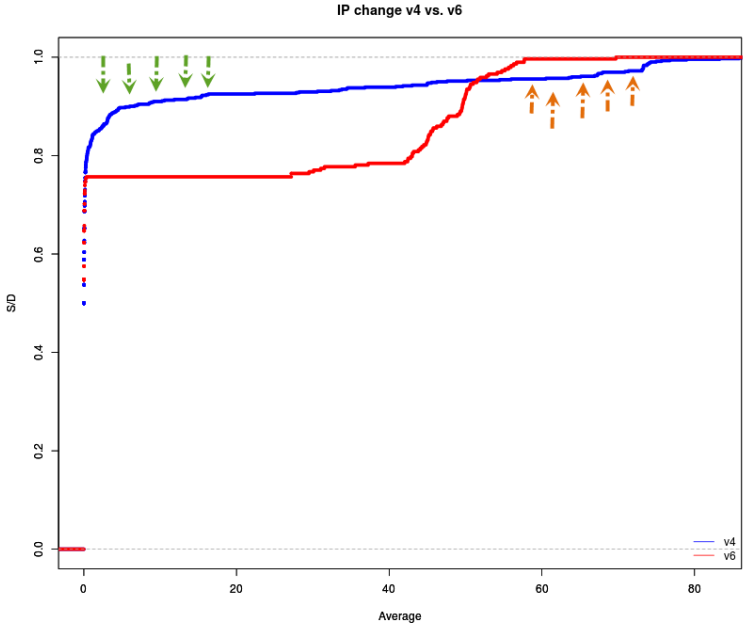
As has been noted before connection over IPv6 from some providers of some NorNet Core site is not available, nevertheless having a consideration on the distribution of source-destination pairs on the Internet path changes is valuable also in this category. In addition investigating the behaviour of Internet path changes and finding some similarity aspect in the additional connections over IPv4 source-destination pair can provide an improvement in the study of the Internet path changes and the real reason of those changes.

In this section the *path changes average per day* results is illustrated for the all source-destination pairs connections regardless of availability of IPv6 connections which has been plotted in *(CDF)*.

As it is observable from the the first plot in figure 5.3 which demonstrates the distribution of source-destination pairs on the Hop based Internet path changes, that the IPv4 connections are less stable than the IPv4 connections which has been shown in the CDF plot of Hop changes in figure 5.2. The additional IPv4 connections caused a difference between these two plots.

It is shown in this plot that about *12%* of IPv4 connections have more than 5 change per day while it is just *7%* for IPv6 connections. Also it is observable that this *12%* of source-destination pairs are distributed exactly



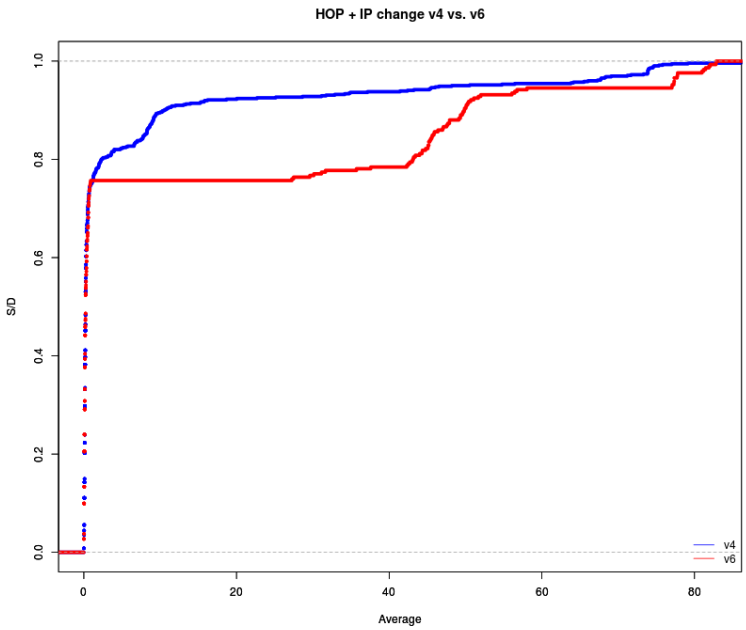




Figure 5.3: Internet path change average per day of all sites over IPv4 and IPv6.

between 5 and 15 hop changes per day (the part that has been marked by the flashes signs in first plot of figure 5.3), which are dominantly connections from different sites of NorNet Core to the *China* site, as well as connections from *Telenor* service providers of *Tromsø* site to the other sites of NorNet Core testbed. It should be noted that the connection is only available over IPv4 for the *China* site and that *Telenor* service provider of *Tromsø* site, which differentiate the Hop change plot of figures 5.2 from 5.3.

The second CDF plot in figure 5.3 displays the distribution of source-destination pairs on the Internet path changes which are based on the IP changes class. As it is observable there are some path instability over IPv4 connections which is around *10%* of source-destination pairs have been distributed between 2 and 20 IP changes (has been shown by *green* flashes). Hence a closer look at the data has revealed that, the IP based Internet path changes **to** the *China* site and **from** the *China* site are distributed in this area.

Another difference which is noticeable between the IP change plot of figures 5.2 and 5.3 is more instability of IPv4 connection which is approximately *10%* of connections with more than 60 changes per day (*orange* flashes) while it is about just *4%* in IP based changes plot in figure 5.2. The inspections showed that this increase is due to additional IPv4 connections to *Versatel* service provider of the *Germany* site. [1]

The third CDF plot in figure 5.3 illustrates the distribution of all connections over IPv4 and IPv6 on the Internet path distribution which are based on the on the HOP + IP changes. Despite the additional IPv4 connection in this plot we can see that IPv6 have more path changes per day while *35%* of IPv6 connection have more than 35 changes per day. while it is still lower for IPv4 connections with around *10%*, more than 35 changes per day.

**The overall observation of IPv4 paths in figure 5.3**:
We can again see the difference between paths with load balancing and those without. We have, however, a set of source-destination pairs that experience quite many path changes, and it is less clear where to draw the line between load balancing and non-load balancing paths. If we use a threshold of **5** changes per day, we find that *18%* of IPv4 source-destination pairs experience load balancing.

[1] *Germany* site have two Internet service providers which just one of them provides connection over both IPv4 and IPv6, and the other provide just supports IPv4 connections.

## 5.3 Path Change Distribution

The aim in this section is to investigate the temporal characteristics of Internet path changes. We seek to understand whether changes are evenly spread out in time, or if they tend to happen in bursts in only a few days. This analysis will also help us in distinguishing *"real"* path changes (caused by routing changes) from path changes caused by load balancing.

This part of result displays exactly the propagation of Internet path changes by only counting the days that have changes and the other days which have not seen any changes are excluded. It shows the scholar perspective of Internet path changes scattering in the days that we have path changes.

This data processing procedure will be shown in scatter plot in order to easily demonstrating the correlation between the fraction of days that have Internet path changes and the number of changes in those days.

The X-axis in the scatter plot, illustrates the fraction of days that have Internet path changes and the average number of changes has been shown on the Y-axis.

Each dot in the scatter plot demonstrates the connection between one source-destination pair, in the *blue* and *red* dots for connections over IPv4 and IPv6 respectively. Hence each dot in the plot shows the correlation of the average number of Internet path changes and the days that have changes in the connection between a single source-destination pair. In this way one dot illustrate a single source-destination pair which have $\alpha$ Internet path changes (on the Y-axis) in the $\beta$ fraction of days that have changes (on the X-axis).

This type of data processing provides a categorization of source-destination pairs by considering the amount of Internet path changes in the change days. If we divide the platform of the plot into four section, Each part will demonstrates different types of distributions as follow:

- **Left-down** corner of the plot illustrates the distribution of source-destination pairs which are experienced Internet path changes in few days of the measurement time period and also few number of changes in those days.

- **Left-top** corner of the platform which shows the distribution of source-destination pairs with many Internet path changes in few days of the measurement time period or in other word shows a burst of path changes in few days. It means that those source-destination pairs do not faced path changes every day but they had many changes just in very few days. The Internet path between those source-destination pairs are change approximately in every it-

eration of path measurement, while we observe only very few source-destination pairs in this category. The number of repetition of Internet measurement path service *per hour* and *per day* can be found in table 4.2.

- **Right-down** corner illustrates Internet path changes of source-destination pairs which have quite few changes per day but they have path changes almost every day.

- **Right-top** corner proposes the distribution of source-destination pairs which have many Internet path changes and almost every day. The Internet path between these source-destination pairs are changed more frequently and they are experienced path change almost in each single iteration of measurement in every days.

As has been noted before, due to the lack of connections over IPv6 for all NorNet Core sites and inequality of the number of IPv4 and IPv6 connections, this data processing part also like the previous one will be split into two sections as follows:

### 5.3.1 Path change distribution of source-destination pairs for IPv6 available sites

In this section the Internet path change distribution only in the days that have changes is illustrated for the source-destination pairs which connection over both IPv4 and IPv6 has been provided for them, that provides equal connections over IPv4 and IPv6 for a fair comparative study.

In order to see the Internet path changes distribution based on the *Hop changes*, *IP changes* and *Hop + IP changes*, the scatter plot for each one has been shown in figure 5.4.

The first graph in the figure 5.4 proposes the distribution of the Internet path changes which are based on the Hop changes in the days that have changes in the whole measurement time period. As it is observable that the Internet path changes in that plot are scattered mostly on the *left-bottom* corner. It has been noted before that the dots which are distributed in the *left-bottom* corner illustrate the source-destination pairs which have few changes in the days that have changes, and in addition they experienced changes in the few days of the measuring time.

As can be seen from Hop change based scatter plot there are some source-destination pairs with zero(*0*) changes both over IPv4 and IPv6 connections. The deep investigation on the data showed that **6** and **8** source-destination pair connections over Ipv4 and IPv6 respectively have zero

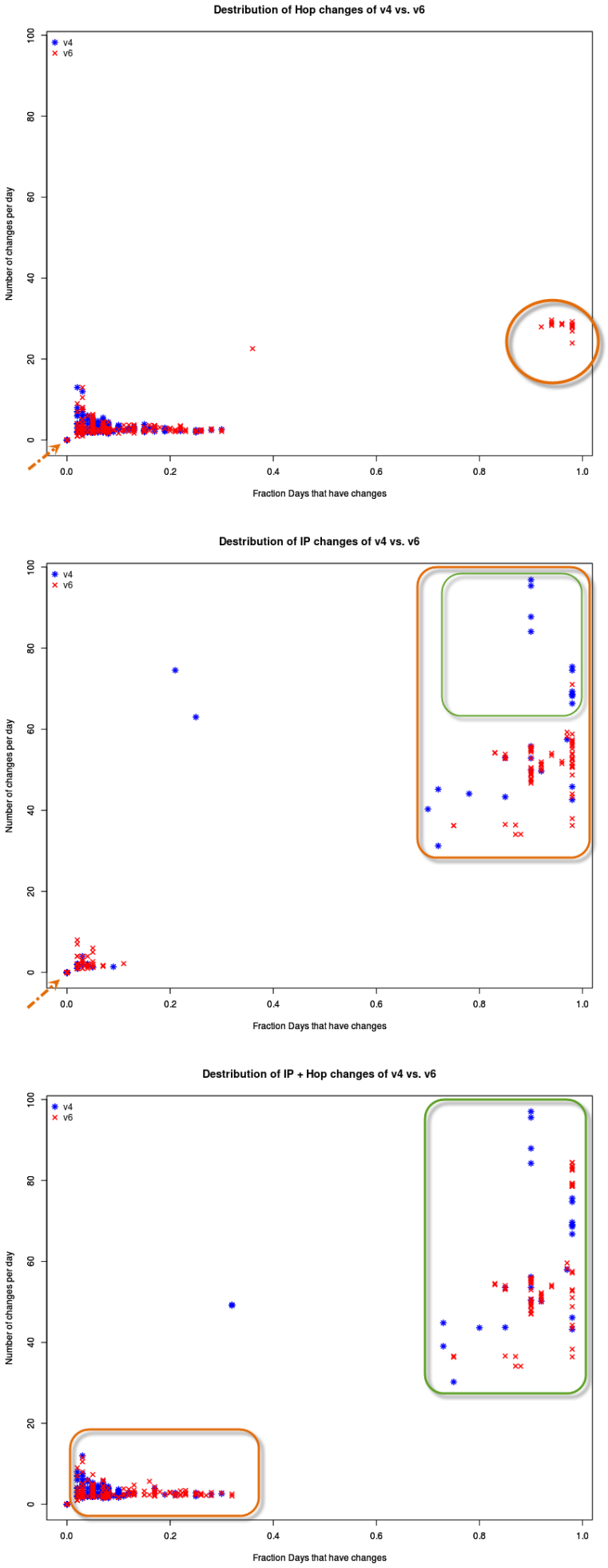



Figure 5.4: Internet path change distribution in change days of only IPV6 available sites over IPv4 and IPv6.

changes. In addition it has been seen that in those zero changes source-destination pairs the source ISP and destination ISP are the same. The similarity of the source and destination ISPs in those zero changes pairs has been seen for both IPv4 and IPv6 connections.

Furthermore as has been noted there are quite few source-destinations that have zero change (6 S/D connections over IPv4, and 8 6 S/D connections over IPv6 ), therefore most of source-destination pairs have experienced some Hop based changes which are ***286*** and ***284*** connections over IPv4 and IPv6 respectively.

As it is visible that all of IPv4 connections are distributed in the *left-bottom* corner with few changes in few days of measuring time that have change. These connections over IPv4 have had between 1 to 10 changes per day only in less than *25%* of total days.

On top of all it is shown that some IPv6 connection have almost frequently changes with more than 20 changes per *95%* of days [2]. More Deeper studies on those IPv6 connection showed us that these are connections from one of Internet service providers of *Simula research laboratory* site to some other IPv6 destinations. Some research on that specific source-destination pairs showed us that there is a routing problem (routing loop) in the *Simula* site for that *Uninett* Internet provider that caused this behaviour.

The second scatter plot in figure 5.4 illustrates the distribution of Internet path changes which are based on the IP changes, in the days that there were changes. As it is visible from the plot the path changes are distributed in the *left-bottom* corner which shows few changes in few days and *right-top* corner which shows many changes in many days of the measuring time period.

The number of source-destination pairs with zero(*0*) changes in IP based changes are more than Hop base change one. There are **228** pairs over IPv4 and **160** pairs over IPv6. It means that overall number of IP based path change is lower than Hop base path change, moreover it shows that source-destination pair connections over IPv4 experience less IP base Internet path change than connections over IPv6.

It is possible to observe from IP based changes plot that some source-destination pair connections over both IPv4 and IPv6 have Internet path changes more than *75%* of days with more than *35* changes per day. Profound studies on IPv4 source-destination pairs on that area [3] showed us

[2]The area that has been shown with orange circle frame in the Hop change plot is connections over IPv6 from Simula research laboratory site.

[3]The area which has been shown with orange rectangle frame in the IP based changes plot are connections over IPv4 with having one of ISP of Germany site as source or destination IP address.

a similarity between each source-destination on that area. Those source-destination pairs are connections **From** and **To** *DFN* Internet service provider in *Germany* site.

In addition it can be seen from this plot that some of those connections between *Germany* site and other sites over IPv4 have almost every day changes (*90%-100%*), with approximately more than 70 changes per each days [4].

This much changes per day which is around half or one-third of whole running measurement service iteration on each day (Number of measurement iteration can be seen in table 4.2.), can prove the existence of load balancer in the path of those source-destination pairs.

The third scatter plot in the figure 5.4 shows the distribution of Internet path changes of source-destination pairs which are based on the IP and Hop change class, in the days that have changes. As we can see that the all source-destination pairs has been divided in two groups. The first group which has been shown in *orange* rectangle in the *left-bottom* corner of the plot platform, demonstrates the source-destination pairs that have few changes with less than 20 changes per day just in less than *30%* of days. This behaviour of the connections between these source-destination pairs can be happen due to the routing problem or routing reconfiguration of one or more intermediate routers in their path which cause this instability. The infrequently changes of these source-destination pairs is a sign of routing change as the reason of that the routing changes are happening so often and time to time, however from here we can not say these changes are real path changes, further analysis are needed which will be proposed later.

The second group in the *right-top* corner in the Hop + IP change scatter plot in the figure 5.4, which is showed by *green* rectangle, illustrates the distribution of source-destination pairs which have many changes with more than 40 changes per day in many days of measurement days with approximately *75%* of days.

The real path changes which are caused from the routing problem or routing reconfiguration usually happening not frequently, thus this kind of behaviour of this group are more likely to happen due to the load balancing which transfers packets to more than one specific router. The frequently path changes in the *right-top* corner is more than one-third of whole measurement iterations in a single day. In addition they have changes in more than *75%* of days.

We found out that there are **262** out of 292 source-destination pairs which connected over IPv4 and **213** out of 292 connections over IPv6 are distributed in the *left-bottom* corner. Further more the distribution of source-destination pairs over IPv4 and IPv6 in the *right-top* corner are **24** and **71** respectively. It should be mentioned **6** and **8** source-destination

[4]This area has been shown in green rectangle frame in the IP based changes plot.

pairs over IPv4 and IPv6 respectively, have experience zero path changes.

### 5.3.2 Internet path change distribution measured for all source-destination pairs

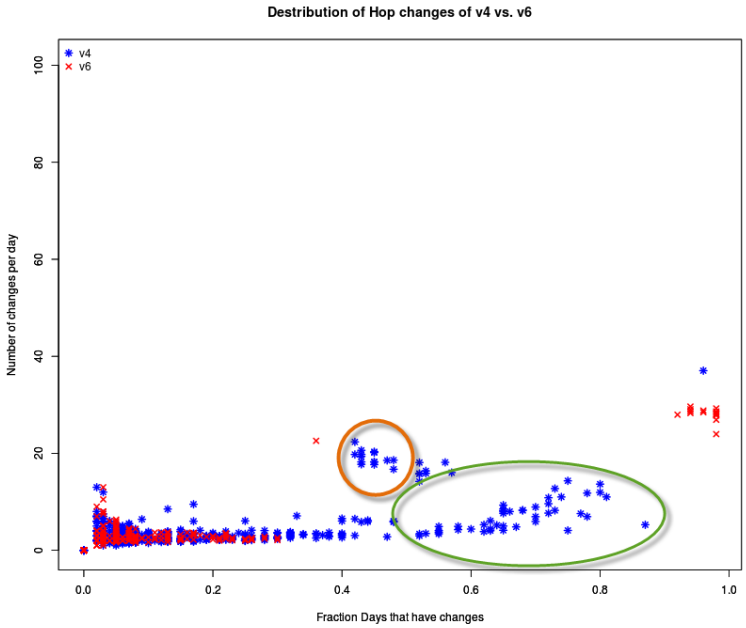
In this section the distribution of Internet path changes of connections between all source-destination pairs over IPv4 and IPv6 on the days of measurement that have changes. The following plots are illustrates how all source-destination pairs behave in the matter of path changes while the IPv6 distribution is the same of the previous part but some additional IPv4 source-destination pairs has been added which has been excluded in the previous part. These additional IPv4 connections are due to the lack of IPv6 availability from some Internet service providers of several sites. However studying the stability of these additional IPv4 connection individually is valuable, specially viewing the stability or instability of connections between some long distances site.

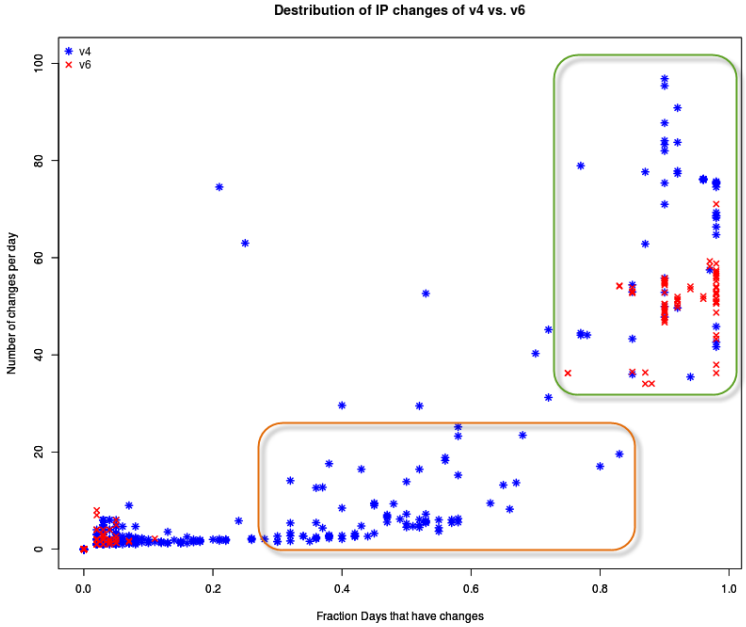
This type of data processing also has been done for different class of Internet path change such as *Hop change*, *IP change* and *Hop + IP change* as has been demonstrated in figure 5.5.

First scatter plot in figure 5.5 shows the distribution of Internet path changes of each source-destination pairs which are based on Hop change class on the days that have changes. By having a glance on the graph the distribution of additional IPv4 connection is easily recognizable. It is possible to see the difference of IPv4 distribution on Hop change plot of figure 5.5 and the Hop change plot of the figure 5.4 while the additional IPv4 source-destination are distributed throughout the X-axis.

As it is observable from the Hop change scatter plot in figure 5.4 the IPv4 connections have less than 22 Internet path changes per day in average, which some of them are distributed on the *left-bottom* corner hence those source-destination pairs have few changes in few days as has been explained before. Whereas the interesting part is the distribution of IPv4 connection on the *right-bottom* corner which have not seen the same behaviour on the previous Hop change plot of the figure 5.4.

This part which has been shown by *green* circle have less than 20 path changes per day based on Hop changes, in more than *50%* of the days. Deep investigation on these source-destination pairs showed that they are connections with destination IP address of *China* site, which means these are connections from other sites of NorNet Core to the *china* site. On top that the studies on the connections between other sites and *China* site showed us that the Hop change of connections over IPv4 from *China* site to the other sites are distributed between *10%* and *45%* of days which located in the introduced *left-bottom* corner. Since Hop changes of the reverse connection from all other site to the *China* site are distributed between



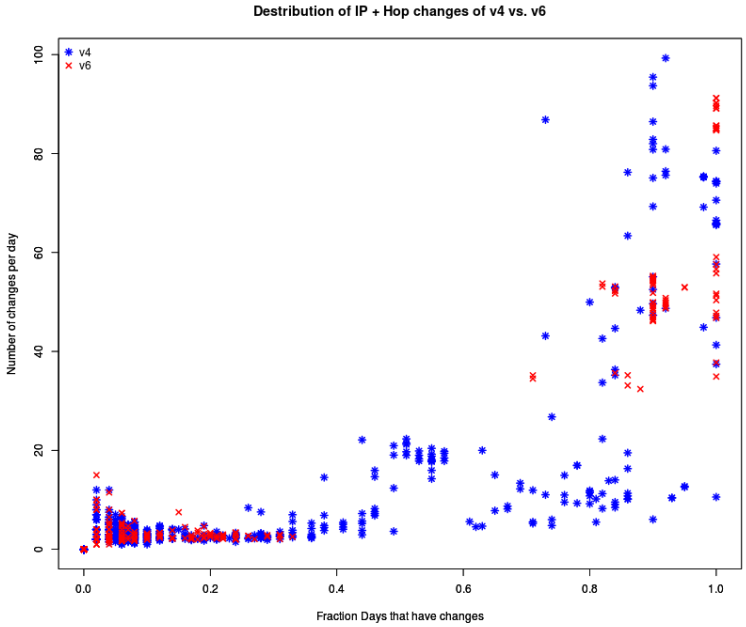




Figure 5.5: Internet path change distribution in change days of all sites over IPv4 and IPv6.

between *50%* and *90%* of days which are located on the *right-bottom* corner of the plot.

In other word paths from *Europe* to *china* have been experience more changes than those from *China* to *Europe*.

In addition the *orange* circle area illustrates the connections over IPv4 from *Telenor* Internet service provider of *Tromsø* site to the other site which have around 20 changes per day in *45%* of days.

The exact reason of this kind of behaviour of these source-destination pairs which are belong the mentioned area*(green and orange circles)* is not so much clear while they are not belong to the defined section of the platform which has been introduced as real path changes or load balancing. However according to the *green* area which are connections to the *China* site, showed the path changes of one-sixth of whole measurement iteration in each day or in another word the Hop path change happened every six runs of measurement in a day. The long distance of other NorNet Core site that dominantly located in *Norway*, to the *China* site can be the reason of behaviour which can be happen due to the routing change of one of many intermediate routers in these routes. One possible explanation could be some kind of load-dependent load balancing, where load balancing only kicks in when there is much traffic. Digging deeper into this is a subject for further studies.

The source-destination pairs in the *orange* area which are connections from *Telenor* ISP in *Tromsø* site, have a similarities in the service provider. These Hop changes in that area may caused due to the specific configuration of that ISP.

The second scatter plot in the figure 5.5 displays the Internet path changes which are based on the IP Changes on the days that have changes for the all NorNet core sites regardless of corresponding IPv6 connection availability to IPv4 connections in all sites. Thus it shows some additional connections over IPv4 compare to the IP change plot of figure 5.4.

As it is visible from the *orange* rectangle area on the plot, there are some connections between source-destination pairs over IPv4 which distributed between *35%* and *90%* of days and just with few changes per days which is less than 20 IP change per those days. The studies on these source-destination connections showed us these connections all come from different NorNet Core site to the *China* site. Like the Hop change plot the connections from *China* site to other sites are distributed at the *left-bottom* corner while the connections to *China* sites are distributed on *right-bottom* site.

The distinction of connections **from** *China* site and **to** *China* site can be sign of path asymmetry which shows the difference between the routes from *China* site to a specific destinations and the reverse route from that

destination to the *China* site.

The area on the IP change scatter plot which has been shown by *green* rectangle demonstrates the distribution of source-destination pairs which have many changes in many days. The *blue* dots shows the IP path changes of connection over IPv4 from and to *Germany* site thus by comparing with the IP change plot in the previous figure 5.4, the increase in the number *blue* dots is visible. This increase is due to the additional IPv4 connections of *Germany* site which has been provided by lack of IPv6 connectivity in *Versatel* service provider of *Germany* site. It means the connections over IPv4 between *Germany* site and some other site of NorNet core testbed has been experienced IP changes in more than *80%* of days and more than 60 changes oer day.

The third scatter plot in figure 5.5 shows the Internet path change distribution which are based on the Hop + IP changes, on the days that have change for all sites of NorNet Core testbed. As has been explained the additional IPv4 connection provide the difference between plot in figure 5.5 and 5.4.

## 5.4 Real Internet Path Changes

The observation from *average of Internet path changes(CDF plot)* in figure 5.2 and *distribution of Internet path changes(scatter plot)* in figure 5.4 proved that there are more *load balancer* in IPv6 connections but did not illustrated the real path changes which have been happened due to the routing configuration changes. In order to extract the *real path changes*, it is necessary to extract the source-destination pairs which have been distributed in the *left-bottom* corner of scatter plot in figure 5.4 and study the average of Internet path changes per of those source-destination pairs over IPv4 and IPv6.

The Internet path changes average per day over IPv4 and IPv6 has been illustrated in figure 5.6 while *blue* lines shows the IPv4 connections and *red* line shows the IPv6 connections for the equal number connection over IPv4 and IPv6. The *green* line shows the few real path changes of IPv4 connections of all sites

As it is visible from CDF plot in figure 5.6 the IPv6 source-destination pairs are less stable than IPv4 source-destination pairs, while *20%* of IPv6 source-destination pairs have more than *0.4* path changes per day in average, hence this is just *2%* for IPv4 source-destination pairs.

Digging into data in database for just this section of few path changes showed that IPv4 paths have in total ***0.16*** path changes per day, while IPv6 paths have ***0.23*** changes per day in average. From this, we conclude that IPv6 paths are less stable than IPv4 paths between the same set of end

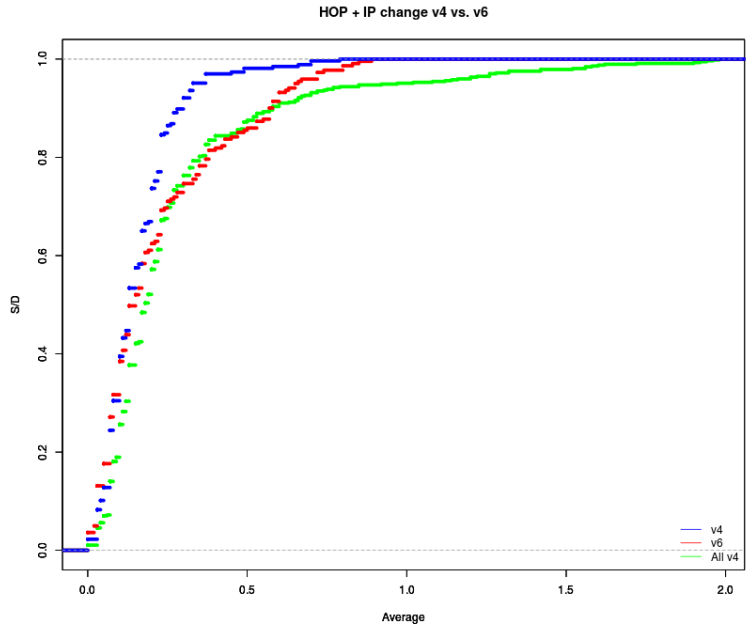



Figure 5.6: Internet path average per change days of only IPV6 available sites over IPv4 and IPv6.

nodes.

Due to the additional IPv4 source-destination pairs in *green* line we can see more instability on that, while the total average of number of changes per day for these non-load balancing source-destination pairs is ***0.28***. As noted before, this set of IPv4 source-destination pairs contains hosts in remote locations (notably *China* and *Germany*), as well as a higher fraction of SD pairs on **ADSL** connections. We believe these differences cause the higher change rate than observed in limited IPv4 connections in *blue* line.

## 5.5 Grouping Internet Path Change into events

In this part the processing of grouping the Internet path changes into event based on the 1 hour threshold and illustrating the distribution of events based on their size has been explained. Here the Internet path change which are measured based on the *Hop changes*, *IP changes* and *Hop + IP changes*, has been divided to the events. The events has been created by having a time difference between them more than 1 hour of threshold time. Hence each events have one or more path change which we know it as event size.

The distribution of source-destination pairs over IPv4 and IPv6 on the different event size has been illustrated by *Probability Mass Function*(**PMF**) which follows formula 5.2 while connections over IPv4 are showed by *blue*

line and IPv6 connections are showed by *red* line. In addition the *Cumulative Density Function*(**CDF**) has been proposed for each correlated PMF plot which provides an informative knowledge about the distribution of events of each source-destination pairs on the events size. The CDF follows formula 5.1.

The x-axis on the both CDF and PMF plot illustrates the events size or number of Internet path changes of the events and the Y-axis shows the density of events of source-destination pairs on different events size.

This data processing step provides a classification of Internet path changes by splitting them into *small events* and *large event* based on their size. The small events illustrates the source-destination pairs with infrequently Internet path changes while the long events shows the source-destination pairs with frequently Internet path changes.

This section is split into three subsections which demonstrated the distribution of events of connections over IPv4 and IPv6 which has been created based on *Hlop change*, *IP change* and *Hop + IP changes*, for equal number of source-destination pairs which have connectivity both over IPv4 and IPv6 as follow.

### 5.5.1 Hop Internet path change events

The plots in figure 5.7 shows the distribution of events based on the Hop Internet path change in PMF and CDF respectively. As it is observable from Hop change plots the events in connections between source-destination pairs over IPv4 have more small events compare to the events of IPv6 connections. The area which has been specified by *orange* circle proposes approximately *85%* of the events of source-destination connections over IPv4 have less than *4* path changes in each event while it is around *57%* for IPv6 events.

The PMF plot in figure 5.7 has been separated into two part. Both IPv4 and IPv6 event curves are divided into the two section which small size events are located in the *right* section of plot and the tail of the curves shows the large size events.

As we can see from the CDF plot of 5.7 which is based on Hop changes, IPv6 events increased slowly and continued till 45 event size while IPv4 had a quick increase and continued till 15 event size. It is visible from the the *green* circle on the PMF plot of figure 5.7 the more percentage of IPv6 events distributed in more than *6* event size than Ipv4 events. The CDF plot of Hop change in figure 5.7 shows that just *3%* of IPv4 events have more than *6* event size while it is *23%* for IPv6 events. Therefore there are more IPv6 events which have large size events that means the connections over IPv6 have regular path changes which are shorter than 1 hour time difference between each.

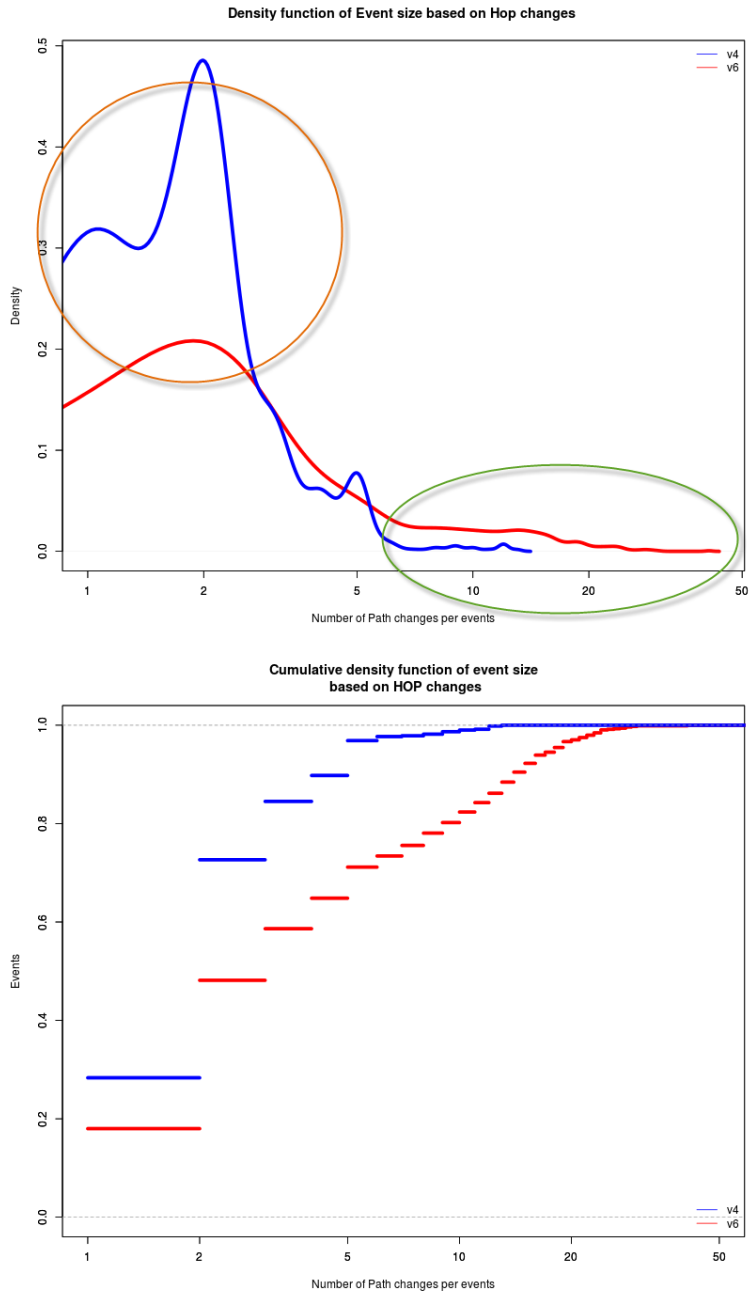



Figure 5.7: Event distribution on event size for only IPV6 available sites over IPv4 and IPv6.

### 5.5.2 IP Internet path change events

The plots in figure 5.8 illustrates the distribution of events of connections between source-destination pairs over IPv4 and IPv6 in the different events size which are based on IP Internet path changes. The first plot shows the *probability mass function(PMF)* of the event distribution and the second one shows in the *cumulative density function(CDF)*.

As it is visible from the plots in 5.8, the created event out of IP changes of each source-destination pairs have larger event size both over IPv4 and IPv6 connections compare to the Hop base changes plot in figure 5.7 while we can see the events with size more than 500 IP path change.

As we can see the difference between small size events between IPv4 connections and IPv6 connections is small while there are around *20%* of IPv4 events which have less than *5* event size while it is around *15%* of events for IPv6. Hence there are still more IPv4 small events than IPv6 small events.

As can be seen from PMF and CDF plots of figure 5.8, Near *50%* of both IPv4 and IPv6 events are distributed between *10* and *50* event size.

In addition the area in CDF and PMF plots which has been shown in *orange* circle shows that the approximately *30%* of IPv4 events are larger than *50* event size while it is just *3%* of events that are distributed in more than *50* event size. It shows that in the IP path change class the frequency of IPv4 path changes are more than IPv6 which created longer events with larger event size.

### 5.5.3 Hop + IP Internet path change events

In this section the distribution of created events in the connections between source-destination pairs on the different event size has been demonstrated. The events has bee created based on both Hop and IP changes, over IPv4 and IPv6. The plots in figure 5.9 illustrated the event distribution in PMF and CDF format.

As has been mentioned before that this part illustrate the distribution of events of equal connections over IPv4 and IPv6 on the different event size, furthermore for both Hop + IP Internet path change which proposes the overall classification of Internet path changes by grouping them into events.

As we can see from the PMF plot in figure 5.9 the density of small size event of IPv4 connections is higher than IPv6 connection, Hence the CDF plot shows that *60%* of IPv4 events have less than *10* event size, and it is

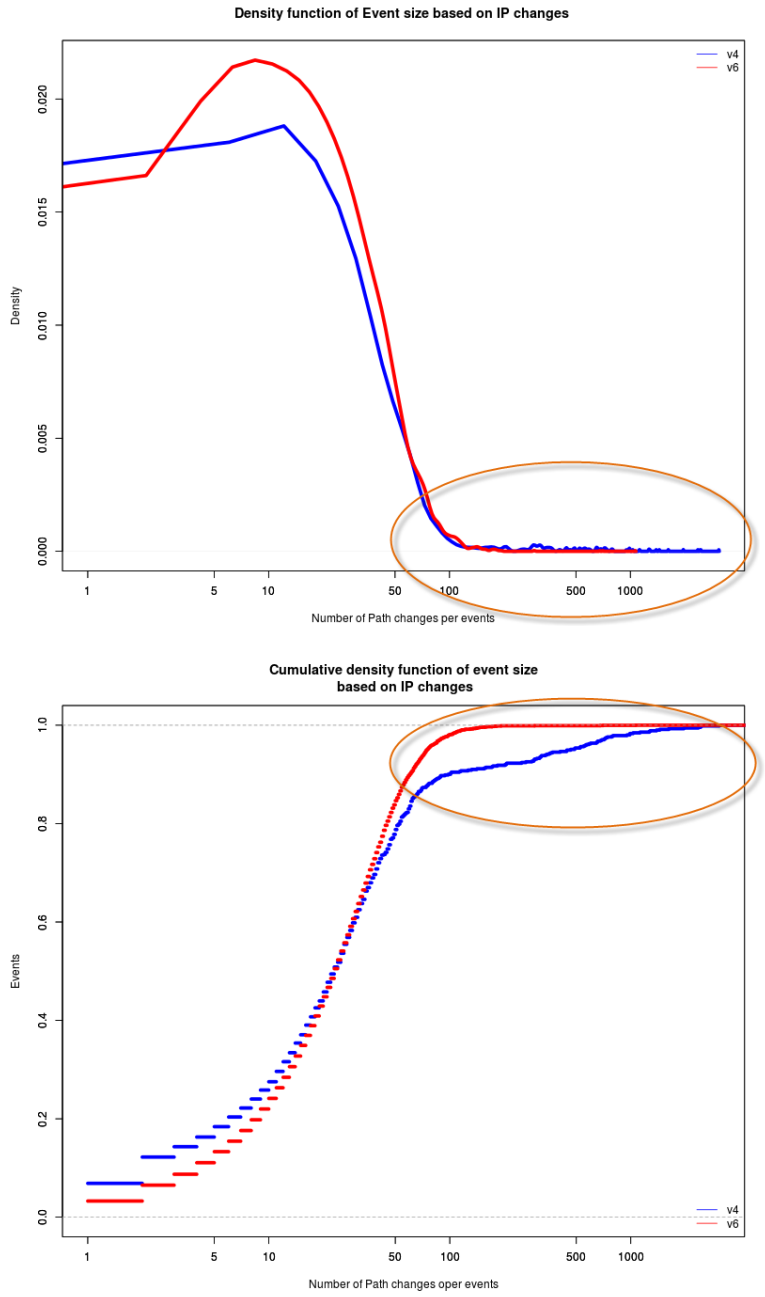



Figure 5.8: Event distribution on event size for only IPV6 available sites over IPv4 and IPv6.

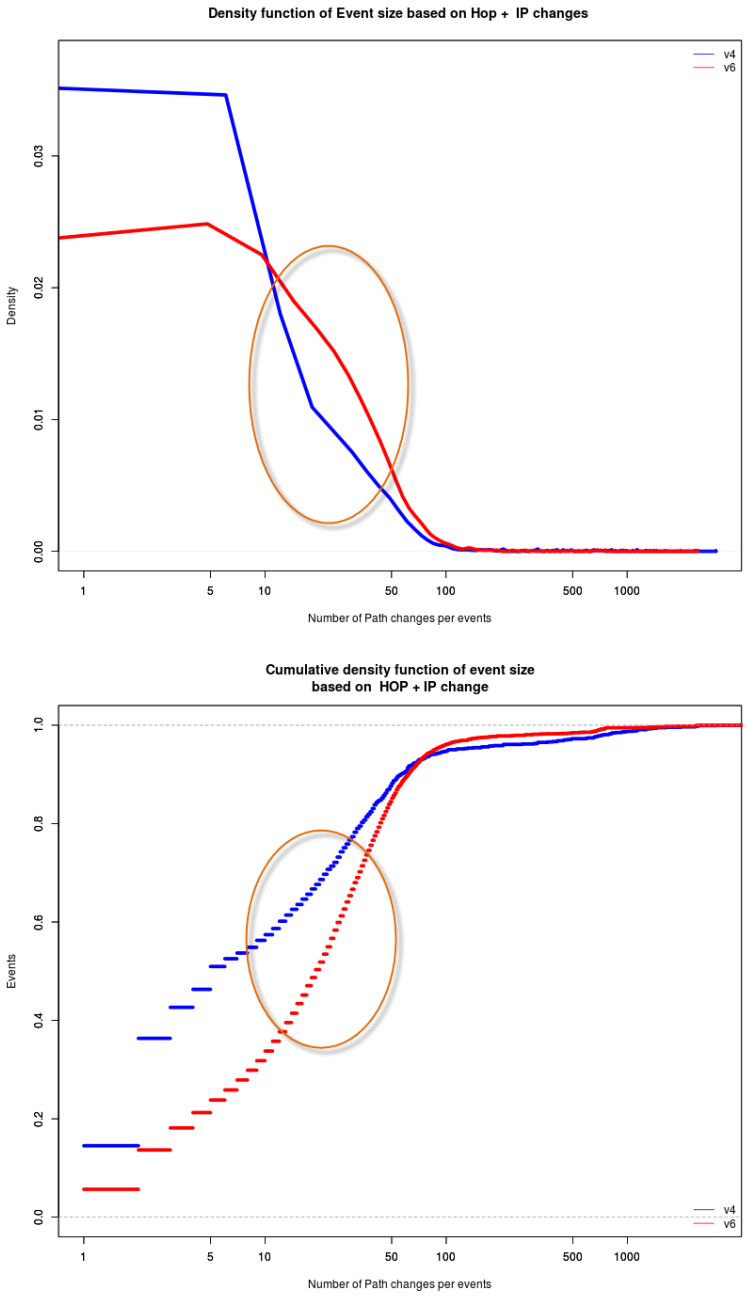



Figure 5.9: Event distribution on event size for only IPV6 available sites over IPv4 and IPv6.

*40%* for IPv6 event.

On the other hand it is visible from both plots from the areas that has been shown by *orange* circles, that just around *20%* of IPv4 events has been distributed between *10* and *50* event size while it is more for IPv6 events with *40%* of events in that area.

As has been noted before the tail of PMF plot shows the large size events which have been continued for a long time with many Internet path changes. Although both IPv4 and IPv6 events can be seen in the tail of PMF plot but the the percentage of IPV6 large events is more than IPv4 large events.

## 5.6 Summary of Findings

This section provide an overview of the analysis which have been derived from all different data processes, while this overview or in other word symmetrization of analysis would be focus on studies of the equal connection over IPv4 and IPv6 source-destination pairs which provides a fairness comparison.

As we have observed from data processes of *Path Change Average per day, Path Change Distribution* and *Grouping Internet Path Change into events* in sections 5.2, 5.3 and 5.5 respectively, a dichotomy has been shown in the all them that all plots are separated into two parts which one part is source-destination pairs with frequent Internet path changes and the other part is source-destination pairs with few and infrequent changes.

Four different data processing which have performed are as follows:

- **Path length distribution** shows how long are the Internet paths over IPv4 and IPv6 between source-destination pairs which has been shows in figure 5.1
- **Path change average per day** data processing illustrates how source-destinations behave in the matter of Internet path changes. This data processing basically shows the combined of real path changes and load balancing changes and has been demonstrated on figure 5.2.
- **Path change distribution** data processing represents the distribution of source-destination pairs in change days. This data processing has been provided a dichotomy of Internet path changes into source-destination pairs with infrequent changes and frequent changes and has been shown in figure 5.4.
- **Real Internet path changes** data processing has been performed by excluding the load balancing paths and providing a comparison

study on source-destinations pairs which have been experienced few path changes. The real Internet path changes has been demonstrated in **CDF** plot in figure 5.6.

- **Path changes grouping into events** data processing illustrates a classification of Internet path changes between source-destination pairs based on the change frequency which has been shown in figures 5.7,5.8 and 5.9.

The summary of observation is as follow:

1. **Internet path length**

   The Internet path length distribution(**CDF** and **PMf**) plots in figure 5.1 shows that IPv6 connections have longer path than IPv4 connections.

2. **Source-destinations with infrequent changes**

   **IPv4** connections between source-destination pairs has been shown in *blue* line or points in plots. The average path changes **CDF** plot in figure 5.2, has been illustrated that *90%* have very few changes with less than 20 changes per day.

   The **scatter** plot in figure 5.4 has been presented that most of IPv4 connections are distributed in *left-right* corner which have few changes in few days with less than 20 changes in less than *40%* of change days while the exact number of IPv4 connections in that area is **262** out 292 pairs while is the almost the *90%* of source-destination pairs in the path change average CDF plot(5.2).

   However the *real path changes(**CDF**)* plot in figure 5.6 illustrated that *2%* IPv4 connections have more than *0.4* Internet path changes per day in average. Hence it means the more stability of IPv4 connection than IPv6 connections.

   On top that the events(**CDF** and **PMF**) plots in figure 5.9 have a close relationship with these observation while shows approximate *60%* of IPv4 connections have short and small size events with less than 20 of event size, hence shows the infrequently changes on those source-destination pairs.

   **IPv6** connections between source-destination pairs has been shown in *red* line or points in the plots. The average path changes(**CDF**) plot in figure 5.2 showed that *73%* of source-destination pairs have few changes in average with less than 20 changes per day while the observation from the distribution(**scatter**) plot in figure 5.4 is the

same. It has been showed that **213** IPv6 connections have less than 20 path changes in less than *40%* per change days which is equal to the *73%* value from path change average CDF plot (5.2).

The real path changes which have been extracted and demonstrate in figure 5.6 proved that *20%* IPv6 connections have more than *0.4* changes per day in average which shows less stability than IPv4 connections.

The observation from events(**PMF** and **CDF**) plots in figure 5.9 also have a close relationship with these observations, while around *40%* of IPv6 connection have events less than 20 event size.

3. **Source-destination pairs with frequent changes**

   **IPv4** connections in path change average(**CDF**) plot in figure 5.2 shows that just *8%* of source-destination pairs have more than 40 changes per day which have a close relationship with the path change distribution(**scatter**) plot in figure 5.4 while 24 source-destination pairs over IPv4 connection have been distributed in*right-top* cornet which shows many changes in many change days, while it is almost the same as the *8%* of IPv4 connection in path changes average CDF plot in figure 5.2.

   The events distribution(**PMF** and **CDF**) plot shows the same observation while *20%* of IPv4 connection events have more than 40 event size.

   **IPv6** source-destination pairs in path change average(**CDF**) plot have been demonstrated that *24%* of them have more than 40 path changes per day. Hence is the same observation from Internet path change distribution(**scatter**) plot in figure 5.4 while shows 71 IPv6 connection have been distributed in *right-top* corner with more than 40 changes in more than *30%* of change days that is the almost the same as *24%* from the path change average CDF plot in 5.2.

   Close relationship have been seen from the events distribution(**PMF** and **CDF**) plots in figure 5.9 while shows *40%* of IPv6 events have more than 40 event size.

# Chapter 6

# Discussion and Future Work

Measuring the Internet path changes and providing a comparison study in the set of IPv4 and IPv6 Internet path changes, as well as distinguishing the real path changes from the load balancing which contributed to the analysis of nature and reason of Internet path changes was the aim of this thesis. The design of this project which led into the service implementation, beside the extracted result and analysis have been presented in the previous chapters. In this chapter the Internet path measurement service implementation, practical result and analysis from the previous chapter would be discussed. Any potential weaknesses of the measurement service, possible modification and improvement and finally future works would be discuss as well.

## 6.1 Overview on the implementation

This section provide an overview of the chosen Internet path measurement structure related to the problem statement.

### 6.1.1 Active, long-term measurement

According to the measuring the stability of end-to-end Internet paths over IPv4 and IPv6 which is the first aim of this thesis, a measurement service was needed that can measure the Internet path over both Internet protocols actively and for a long time. In this fashion the *NorNet-Trace* service has been developed which runs as a service in all NorNet Core sites and captures the Internet path from all interface of local site to all interfaces of all remote sites over IPv4 and IPv6.

The measurements service works by sending *ping(ICMP)* packets from source IP to the remote IP by increasing *TTL* values and get the IP address of intermediate hops back which create a path by the sequence of IP address of intermediate routers between source and destination nodes.

In addition the *Round-Trip-Time(RTT)* of hop-by hop and end-to-end are measured by the *NorNet-Trace* service. The *NorNet-Trace* service has been

executed and captured Internet paths continuously for a complete 2 month.

In the next stage all measured values are inserted into the database in a centralizing way which the captured data from all sites are transferred to the database time to time.

### 6.1.2 Path changes extraction and data processes

The inserted Internet paths over IPv4 and IPv6 between each source-destination pairs are needed to be process in order to extract the path changes. In this way the Internet path changes has been divided into three classes of *Hop changes*, *IP changes* and *Star changes* while the decision has been made to escaping the *Star changes* class while the *Stars* in the paths are more likely to be due to the limitation of some of routers to answer to the *ICMP* packets.

The data processes on the path changes have been split into two groups, One for the equal number of source-destination pairs that have connectivity over IPv4 and IPv6 which has been used for the comparison study of the behaviour of IPv4 and IPv6 connections in the stability perspective. Second group is all source-destination pairs regardless of equality of IPv4 and IPv6 connection, while it illustrated the valuable behaviour of additional IPv4 connection specially for the long distance connections.

***Internet path length distribution*** over IPv4 and IPv6 was one of the data processing while demonstrated that the connections over IPv6 are relatively longer than IPv4 connections between source-destination pairs.

***Internet path changes average per day*** was another data processing that has been performed. The distribution of Internet path changes of the different class of *Hop changes*, *IP changes* and also *Hop + IP changes* in the connections between source-destination pairs over IPv4 and IPv6, on the total measurement days has been processed. This data processing showed that the IPv4 connection have less Internet path changes than IPv6 connections.

***Distribution of Internet path changes on the day that have changes*** was the third performed data processing. This data processing type illustrated that the source-destination pairs are distributed in two parts. One group of source-destination pairs have few changes in few days, and the other group have many changes in many changes. It has been measured also from that data processing that more IPv4 connections have few changes in few days, while more IPv6 connection have many changes in many changes.

It has been analyzed that source-destination pairs with many changes in many days are cause due to the load balancing and are not real path changes. On the other hand the source-destination pairs with few path changes in few days, did not illustrates the real path changes which caused

from routing configuration changes.

***Real path changes*** have been extracted from the source-destination pairs which have few Internet path changes in few days and illustrates the average of Internet path changes of each source-destination pairs per day in the CDF plot of figure 5.6.

From that plot we can see that IPv6 source-destination pairs have more Internet path changes in average which proved the less stability of IPv6 connections than IPv4 connections.

***Internet path changes grouping into events*** was the fourth data processing which has been done. This data processing provide a classification Internet path changes of source-destination pairs in the short time events with low size which few path changes are included in them and the changes has been happened infrequently , and long time event with large event size which path changes has been done more frequently.

It has been shown in that data processing type that IPv4 connections have more short events while IPv6 connections have more longer events. In addition the analysis have been proposed that the long time events which have frequent path changes are caused from load balancing while short time vents that have rare path changes are real path changes.

## 6.2 Usability of the service

This service and measurement can be useful in two ways among providing the comparative study of Internet stability over IPv4 and IPv6 as follows:

### 6.2.1 *NorNet Core* testbed experimenter

The measured Internet paths over IPv4 and IPv6 and the stability of each path between source-destination pairs of *NorNet Core* testbed is valuable for the experimenters of the *NorNet Core* testbed. The persons who conducting experiments by using *NorNet Core* testbed infrastructure can use these information which has been derived from this service and choose the better source-destinations pairs which are more stable and fits better their experiments, in order to have more reliable result with better performance.

### 6.2.2 Similar measurement in other networks and testbeds

The similar work can be done in other networks and testbeds in order to get the stability of Internet paths of the network. This service can be used in other networks which can provide the informative knowledge in this fashion.

## 6.3 Potential Weaknesses and Possible Modification

In this section the potential pitfalls of the measurement service in addition to the possible modification which can improve the the reliability and robustness of Internet path measurement would be discussed.

### 6.3.1 Possibilities of false links

***False links*** can be derived from the classic Internet route measurement utilities such as *traceroute and ping* which are suffers from the existence of the load balancing. Load balancers may implies more active routes between source-destination in the Internet route measurements as has been explained in 2.4.2.

In this way it is possible to do some modifications in order to prevent the possible *false link* by using *Paris-traceroute* tool.

### 6.3.2 Using *Paris-traceroute*

The *Paris-traceroute* tool avoids sending the probes through different paths toward destination in the presence of *per flow* load balancer by getting advantages of the header field of probes. We couldn't use the *Paris-traceroute* utility due to the:

*Paris-traceroute* is a valuable utility for the Internet path measurement specially in the presence of load balancers in the routes between a source-destination pair. But *Paris-traceroute* tool at this moment is under development and needs some more improvements, in addition it does not support multiple interface which is an essential feature in our measurement due to the multi homing of NorNet Core sites.

Some efforts has been done for 2 weeks in order to add this feature to the *Paris-traceroute* tool which it was impossible in the scope of this thesis due to the time limitation.

### 6.3.3 Different traffic

In this measurement the Internet paths over IPv4 and IPv6 has been measured just by sending the *Ping(ICMP)* probes from source node to the destination node. Another experiment that could be done was sending different traffic such as *User Datagram protocol(UDP)* or *Transmission Control Protocol(TCP)* and getting the IP address of intermediate back. In this fashion the it is possible to discover different routes between a source-destination pair within the existence of *per-flew* load balancer, while *per-flew* load balancers use the five-tuple(*source address, destination address, protocol, source port, destination port*) of packet header in order to define the forwarding route. Hence by sending different traffic the *protocol* value of packet header five-tuple would be changed and the forwarding flow would be changed.

In addition by sending different traffics, it is probable that the *autonomous* intermediate routers which have been configured to answer to limit

amount of ICMP probes or not answer to ICMP probes at all, reveal.

### 6.3.4 Relationship between the Internet paths delay and stability

AS has been noted before the *hop-by-hop* and *end-to-end* delay has been measured by the *NorNet-Trace* service although it needs some modification and some more systematic approaches for getting the absolute delay at the each moment. Having an analysis on the measured delays of each Internet paths between source-destination pairs could be illustrate a correlation between the stability of the Internet paths and delays of each path.

## 6.4 Future Works

In this section some future works is demonstrated which due to the time limitation could not measured in the scope of this thesis.

### 6.4.1 Internet paths similarities

A measurement experiment and data analysis which could be done to provide the informative knowledge is to answering this question:

***Are the IPv4 and IPv6 paths between a source-destination pair the same?***

This question could be answer in two different perspective as follow:

- **Temporal aspect**
  In our measurement the Internet paths have been measured continuously and in each iteration the exact date and time has been captured, hence the exact time of extracted Internet path changes are known.

  In the conditions that we observe a path change between a source-destination pair over both IPv4 and IPv6, it is possible to study the time of IPv4 path change and IPv6 path changes, thus If the have changes at the same time it is probable that both IPv4 and IPv6 passing the same underlying path.

- **Location aspect**
  In addition to studying the temporal aspect of Internet path changes, it is needed to finding the location of path changes. If we observe a changes in the Internet path over both IPv4 and IPv6 in one of the intermediate routers, if the change of both version of Internet protocol happened in the same hop or machine, it is likely that both IPv4 and IPv4 using same path from source node to the destination node.

#### 6.4.1.1 Methods to quantify Internet path similarities

Several technologies can be used in order to investigate the path similarities which have been explained in following. The usage of following technologies outcome in addition to the measured information in this thesis may led into the result of underlying Internet path similarities between IPv4 and IPv6 connections:

- **Reverse Name resolution or Reverse DNS lookup**

  The detected intermediate hop that has experienced change in the path between a source-destination pair can be recognize by the *reverse DNS lookup* while most of intermediate routers are likely to be assigned a host name beside belonging to a domain in the networks.

  *Reverse DNS lookup* provides a resolution of IP address to the domain name. For instance it shows the name of the machine and it belonging domain name as follow:

Reverse Name Resolution.

```
Server:                127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
218.121.252.85.in-addr.arpa        name = static218.banetele-cust.com.

Authoritative answers can be found from:
121.252.85.in-addr.arpa        nameserver = dns2.as2116.net.
121.252.85.in-addr.arpa        nameserver = dns3.as2116.net.
121.252.85.in-addr.arpa        nameserver = dns1.as2116.net.
dns1.as2116.net        internet address = 193.75.75.1
dns1.as2116.net        has AAAA address 2001:8c0:2001::3:53
dns2.as2116.net        internet address = 193.75.75.4
dns2.as2116.net        has AAAA address 2001:8c0::4
```

  From the above reverse name resolution we can see that the host name(*static218*) and domain name(*banetele-cust.com*) has been appeared.

  In this fashion the hostname and domain name of intermediate hop can be recognize, thereby similarities between the IPv4 and IPv6 intermediate hop in the path that we observe changes on them, will be shown.

- **Geo IP location**

  *Geolocation (Geo IP location)* is a utility the geographical location of intermediate router is deduced which can be used in the matter of investigating the similarities of IPv4 and IPv6 Internet path by recognizing the approximate geographical location of change experienced intermediate hop in the path between a source-destination pair over IPv4 and IPv6.

This tool represents an approximate location of a intermediate router in set of country or city based on its IP address as follow:

Geolocation of IP address.

```
geoiplookup 113.59.104.58
GeoIP Country Edition: CN, China


geoiplookup6 2001:8c0::4
GeoIP Country V6 Edition: NO, Norway
```

The similarity in the approximate geographical location of the intermediate routers in a path between a source-destination pair that have experience change over IPv4 and IPv6 can turn into a conclusion of Internet path similarities in IPv4 and IPv6 path between a source-destination pair.

Others information that can be used in order to help the Internet path similarity are:

- ***Round-Trip-Time* delay**

  A comparative study on the *hop-by-hop* delay and *end-to-end* delay if the IPv4 and IPv6 connection between a source-destination pair can help the recognizing the similarities or dissimilarities of IPv4 and IPv6 underlying Internet paths.

- **Number of Intermediate Hops**

  Number of intermediate routers between a source-destination pair connection over IPv4 and IPv6 can be involve in the comparative study of IPv4 and IPv6 path similarities.

### 6.4.2 Influence of Internet path Instability on end-to-end Performance

Measuring the *end-to-end* performance of the Internet paths of source-destination pairs that have experienced path instability and comparing with the more stable Internet paths can provide informative knowledge about how Internet path stability or instability can influence the performance of connection over both IPv4 and IPv6.

In addition this study can be done for the Internet paths between source-destination pairs that have load balancing and observing whether path load balancers affect the performance of Internet paths or not.

### 6.4.3 Observing the generality of the result

Our actual result and observations may change if we considering some factors:

- Long time measurement.
- More sites or in other word more source-destination pairs.
- Distributed sites. (currently, most of the sites are located in *Norway*, one in *Chine*, one in *Germany*, and one in *Sweden*.
- Diverse Internet service providers.
- More heterogeneous end points.

It is likely that the result changes if we include these factors, hence a future work can be rerunning the whole measurement by adding these factors and observing how the result would be changed.

# Chapter 7

# Conclusion

The main aim of this thesis is to discovering the stability of Internet path over IPv4 and IPv6 and achieve the knowledge of end-to-end Internet path stability by providing the comparison study in the set of IPv4 and IPv6 Internet paths.

At the first stage the measurement service has been developed which captures the Internet paths over IPv4 and IPv6 actively and continuously for a long time.

Some processing has been performed on the data in order to extract the Internet path changes and segregate the real path changes which caused due to the routing configuration changes and the path artificial path changes which caused due to the routes load balancers.

The analysis on the measurement data and Internet path changes presented longer Internet paths in IPv6 than IPv4 which can be derived from fewer links in IPv6 paths which is obviously due to the less deployment of IPv6 from the Internet service providers.

In addition the analysis has been proved that there are more load balancers in IPv6 Internet paths than IPv4 paths. This observation raises a doubt while there is less traffic on IPv6 connection than IPv4 connections. One hypothesis is that a packet which is sent over IPv6 path from source until reaches the destination node is not use a complete underlying IPv6 path through the whole way, rather traversing tunnels in the way to the destination. Other hypothesis is that due to the fewer links on IPv6, the operators has been deployed load balancers in order to distribute current IPv6 traffic loads on different paths in a way that few available links not become overloaded.

Furthermore the analysis proved that there is less stability in IPv6 connections than IPv4 connections. This IPv4 have been deployed and used for many years and the Internet operators have enough experienced about IPv4 deployment and performance, hence it is better managed. Although the IPv6 have been around for many years but it have been not used so

much by the Internet service provider due to the existence and availability of IPv4 address capacity, therefore it was not so much critical for operators to deploy and improve the implementation of IPv6, thus it is less matured and the lack of experience from the Internet service providers is sensible.

# Chapter 8

# Appendix

This chapter contains all scripts that made this thesis work. All the scripts which are written in *Python* are available in repository and can accessible from the following links. The repository can be clone by :

NorNet-Trace repository clone command.

```
git clone https://foroughg@bitbucket.org/foroughg/nornet-trace.git
```

## 8.1 Appendix 1: NorNet-Trace Script

NorNet-Trace Service (http://bit.ly/1o9Gg8q)

## 8.2 Appendix 2: NorNet-Trace-Import script

NorNet-Trace-Import Service (http://bit.ly/1vwnX1G)

## 8.3 Appendix 3: trace-configuration file

trace-configuration file (http://bit.ly/RQGuWH)

## 8.4 Appendix 6: NorNet-Trace-Compare script

NorNet-Trace-Compare Script (http://bit.ly/1p8gpxF)
NorNet-Trace-CDFGraph script (http://bit.ly/1o9G7Cb)

## 8.5 Appendix 4: NorNet-Trace-Distribution Script

NorNet-Trace-Distribution script (http://bit.ly/1ghLcrL)
NorNet-Trace-Scatter script (http://bit.ly/1gJnkxH)

## 8.6 Appendix 5: NorNet-Trace-Event Script

NorNet-Trace-Event script (http://bit.ly/1kkfolq)
NorNet-Trace-CDFEvent script (http://bit.ly/1mQIgTv)

## 8.7 Appendix 7: NorNet-Trace-Length

NorNet-Trace-Length script (http://bit.ly/ToEgiE)

# Bibliography


[1] L. Andersson, I. Minei and B. Thomas. *LDP Specification*. RFC 5036 (Draft Standard). Internet Engineering Task Force, Oct. 2007. URL: http://www.ietf.org/rfc/rfc5036.txt.

[2] *APNIC. APNIC IPv4 Address Pool Reaches Final /8.* Apr. 2011. URL: http://www.apnic.net/publications/news/2011/final-8.

[3] Brice Augustin, Timur Friedman and Renata Teixeira. 'Measuring load-balanced paths in the Internet'. In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM. 2007, pp. 149–160.

[4] Brice Augustin, Timur Friedman and Renata Teixeira. 'Measuring multipath routing in the internet'. In: *IEEE/ACM Transactions on Networking (TON)* 19.3 (2011), pp. 830–840.

[5] Brice Augustin et al. 'Avoiding traceroute anomalies with Paris traceroute'. In: *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM. 2006, pp. 153–158.

[6] Ferdous A Barbhuiya et al. 'An Active Detection Mechanism for Detecting ICMP Based Attacks'. In: *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. IEEE. 2012, pp. 51–58.

[7] R. Bonica et al. *ICMP Extensions for Multiprotocol Label Switching*. RFC 4950 (Proposed Standard). Internet Engineering Task Force, Aug. 2007. URL: http://www.ietf.org/rfc/rfc4950.txt.

[8] S. Branigan et al. 'What can you do with Traceroute?' In: *Internet Computing, IEEE* 5.5 (Sept. 2001), pp. 96–. ISSN: 1089-7801. DOI: 10.1109/4236.957902.

[9] W. Milliken C. Partridge T. Mendez, ed. *RFC 1546, Host Anycasting Service*. Internet Engineering Task Force. Nov. 1993. URL: https://tools.ietf.org/html/rfc1546.

[10] *CAIDA. Archipelago measurement infrastructure*. URL: http://www.caida.org/projects/ark/.

[11] Xiang Chen et al. 'A survey on improving TCP performance over wireless networks'. In: *Resource management in wireless networking*. Springer, 2005, pp. 657–695.

[12] *CISCO: How does load balancing work?* 2005. URL: http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html.

[13] *Configuring Load-Balance Per-Packet Action*. URL: http://www.juniper.net/techpubs/software/junos/junos70/swconfig70-policy/html/policy-actions-config11.html.

[14] A. Conta and S. Deering. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. RFC 2463 (Draft Standard). Obsoleted by RFC 4443. Internet Engineering Task Force, Dec. 1998. URL: http://www.ietf.org/rfc/rfc2463.txt.

[15] S. Deering and R. Hinden. *RFC 2460 Internet Protocol, Version 6 (IPv6) Specification*. Internet Engineering Task Force. Dec. 1998. URL: http://tools.ietf.org/html/rfc2460.

[16] A. Dhamdhere and C. Dovrolis. 'Twelve Years in the Evolution of the Internet Ecosystem'. In: *Networking, IEEE/ACM Transactions on* 19.5 (Oct. 2011), pp. 1420–1433. ISSN: 1063-6692. DOI: 10.1109/TNET.2011.2119327.

[17] Amogh Dhamdhere et al. 'Measuring the Deployment of IPv6: Topology, Routing and Performance'. In: *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*. IMC '12. Boston, Massachusetts, USA: ACM, 2012, pp. 537–550. ISBN: 978-1-4503-1705-4. DOI: 10.1145/2398776.2398832. URL: http://doi.acm.org/10.1145/2398776.2398832.

[18] Benoit Donnet. 'Internet topology discovery'. In: *Data Traffic Monitoring and Analysis*. Springer, 2013, pp. 44–81.

[19] Benoit Donnet and Timur Friedman. 'Internet topology discovery: a survey'. In: *Communications Surveys & Tutorials, IEEE* 9.4 (2007), pp. 56–69.

[20] Benoit Donnet et al. 'Deployment of an algorithm for large-scale topology discovery'. In: *Selected Areas in Communications, IEEE Journal on* 24.12 (2006), pp. 2210–2220.

[21] Benoit Donnet et al. 'Efficient Algorithms for Large-scale Topology Discovery'. In: *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS '05. Banff, Alberta, Canada: ACM, 2005, pp. 327–338. ISBN: 1-59593-022-1. DOI: 10.1145/1064212.1064256. URL: http://doi.acm.org/10.1145/1064212.1064256.

[22] Benoit Donnet et al. 'Revealing MPLS tunnels obscured from traceroute'. In: *ACM SIGCOMM Computer Communication Review* 42.2 (2012), pp. 87–93.

[23] T. Dooley M. ; Rooney, ed. *IPv6 and the Future Internet*. Wiley-IEEE Press: Wiley-IEEE Press, 2013. Chap. 10.

[24] T. Dooley M. ; Rooney, ed. *IPv6 Readiness Assessment*. Wiley-IEEE Press: Wiley-IEEE Press, 2013. Chap. 4.

[25] T. Dreibholz and E.G. Gran. 'Design and Implementation of the NORNET CORE Research Testbed for Multi-homed Systems'. In: *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. Mar. 2013, pp. 1094–1100. DOI: 10.1109/WAINA.2013.71.

[26] Thomas Dreibholz. *An Introduction to NorNet Core for the Site Deployment at Universitetet i Stavanger*. Invited Talk at Universitetet i Stavanger. Stavanger, Rogaland/Norway, 27th Nov. 2012. URL: https://simula.no/publications/Simula.simula.1626/simula_pdf_file.

[27] Thomas Dreibholz. *NorNet – An Open, Large-Scale Testbed for Multi-Homed Systems*. Invited Talk at Swinburne University, Centre for Advanced Internet Architectures. Melbourne, Victoria/Australia, 30th Jan. 2014. URL: https://simula.no/publications/Simula.simula.2488/simula_pdf_file.

[28] E.P. Duarte et al. 'Finding stable cliques of PlanetLab nodes'. In: *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*. June 2010, pp. 317–322. DOI: 10.1109/DSN.2010.5544300.

[29] D. Farinacci et al. *Generic Routing Encapsulation (GRE)*. United States, 2000.

[30] Lixin Gao and J. Rexford. 'Stable Internet routing without global coordination'. In: *Networking, IEEE/ACM Transactions on* 9.6 (Dec. 2001), pp. 681–692. ISSN: 1063-6692. DOI: 10.1109/90.974523.

[31] *Google IPv6 adoption statistic report*. Mar. 2014. URL: http://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption.

[32] Ernst Gunnar Gran, Thomas Dreibholz and Amund Kvalbein. 'NorNet Core – A Multi-Homed Research Testbed'. In: *Computer Networks, Special Issue on Future Internet Testbeds* (3rd Jan. 2014). ISSN 1389-1286. ISSN: 1389-1286. DOI: 10.1016/j.bjp.2013.12.035. URL: https://simula.no/publications/Simula.simula.2236/simula_pdf_file.

[33] Silva Hagen, ed. *IPv6 Essentials*. O'Reilly: O'Reilly Media, 2006.

[34] Yihua He et al. 'On routing asymmetry in the Internet'. In: *Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE*. Vol. 2. IEEE. 2005, 6–pp.

[35] K. Hubbard et al. *Internet Registry IP Allocation Guidelines*. RFC 2050 (Best Current Practice). Internet Engineering Task Force, Nov. 1996. URL: http://www.ietf.org/rfc/rfc2050.txt.

[36] G. HUSTON. 'Address Exhaustion'. In: *The Internet Protocol Journal* 14.1 (2011).

[37] *IPv4 Address Report*. 2009. URL: http://www.potaroo.net/tools/ipv4/index.html.

[38] *IPv6: IPv6 / IPv4 Comparative Statistics*. Mar. 2014. URL: http://bgp.potaroo.net/v6/v6rpt.html.

[39] Ludovic Jacquin et al. 'IBTrack: an ICMP black holes tracker'. In: *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE. 2012, pp. 2827–2833.

[40] S.R. Kamel Tabbakh, B. Mohd Ali and S. Khatun. 'A Novel Router-based Approach for Measuring Packet-based Delay in High Speed IP Networks'. In: *Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on*. Sept. 2010, pp. 119–124. DOI: 10.1109/NETAPPS.2010.29.

[41] Ethan Katz-Bassett et al. 'Reverse traceroute.' In: *NSDI*. Vol. 10. 2010, pp. 219–234.

[42] C. Labovitz, A. Ahuja and F. Jahanian. 'Experimental study of Internet stability and backbone failures'. In: *Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on*. June 1999, pp. 278–285. DOI: 10.1109/FTCS.1999.781062.

[43] C. Labovitz, G.R. Malan and F. Jahanian. 'Internet routing instability'. In: *Networking, IEEE/ACM Transactions on* 6.5 (Oct. 1998), pp. 515–528. ISSN: 1063-6692. DOI: 10.1109/90.731185.

[44] C. Labovitz, G.R. Malan and F. Jahanian. 'Origins of Internet routing instability'. In: *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 1. Mar. 1999, 218–226 vol.1. DOI: 10.1109/INFCOM.1999.749286.

[45] N. Leavitt. 'IPv6: Any Closer to Adoption?' In: *Computer* 44.9 (Sept. 2011), pp. 14–16. ISSN: 0018-9162. DOI: 10.1109/MC.2011.284.

[46] Hongjun Liu et al. 'Locating Routing Instability Based on Path Exploration'. In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*. July 2012, pp. 439–444. DOI: 10.1109/IMIS.2012.47.

[47] Matthew Luckie, Kenjiro Cho and Bill Owens. 'Inferring and Debugging Path MTU Discovery Failures'. In: *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. IMC '05. Berkeley, CA: USENIX Association, 2005, pp. 17–17. URL: http://dl.acm.org/citation.cfm?id=1251086.1251103.

[48] Matthew Luckie, Young Hyun and Bradley Huffaker. 'Traceroute Probe Method and Forward IP Path Inference'. In: *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*. IMC '08. Vouliagmeni, Greece: ACM, 2008, pp. 311–324. ISBN: 978-1-60558-334-1. DOI: 10.1145/1452520.1452557. URL: http://doi.acm.org/10.1145/1452520.1452557.

[49] Matthew Luckie and Ben Stasiewicz. 'Measuring Path MTU Discovery Behaviour'. In: *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. IMC '10. Melbourne, Australia: ACM, 2010, pp. 102–108. ISBN: 978-1-4503-0483-2. DOI: 10.1145/1879141.1879155. URL: http://doi.acm.org/10.1145/1879141.1879155.

[50] Matthew Luckie, Amogh Dhamdhere, David Murrell et al. 'Measured impact of crooked traceroute'. In: *ACM SIGCOMM Computer Communication Review* 41.1 (2011), pp. 14–21.

[51] *M. Toren, Tcptraceroute*. 2001. URL: http://michael.toren.net/code/tcptraceroute/.

[52] G. Malkin. *Traceroute Using an IP Option*. RFC 1393 (Experimental). Internet Engineering Task Force, Jan. 1993. URL: http://www.ietf.org/rfc/rfc1393.txt.

[53] Matthew Mathis et al. 'The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm'. In: *SIGCOMM Comput. Commun. Rev.* 27.3 (July 1997), pp. 67–82. ISSN: 0146-4833. DOI: 10.1145/263932.264023. URL: http://doi.acm.org/10.1145/263932.264023.

[54] N. V. Mnisi, O.J. Oyedapo and A. Kurien. 'Active Throughput Estimation Using RTT of Differing ICMP Packet Sizes'. In: *Broadband Communications, Information Technology Biomedical Applications, 2008 Third International Conference on*. Nov. 2008, pp. 480–485. DOI: 10.1109/BROADCOM.2008.76.

[55] J.C. Mogul and S.E. Deering. *Path MTU discovery*. RFC 1191 (Draft Standard). Internet Engineering Task Force, Nov. 1990. URL: http://www.ietf.org/rfc/rfc1191.txt.

[56] Mike Muuss. *The Story of the PING Program*. Dec. 1983. URL: http://ftp.arl.army.mil/~mike/ping.html.

[57] T. Narten et al. *Neighbor Discovery for IP version 6 (IPv6)*. RFC 4861 (Draft Standard). Updated by RFC 5942. Internet Engineering Task Force, Sept. 2007. URL: http://www.ietf.org/rfc/rfc4861.txt.

[58] *OneLab, PlanetLab Central API Documentation*. 2012. URL: https://www.planet-lab.eu/planetlab/doc/PLCAPI.pdf.

[59] Vern Edward Paxson. 'Measurements and analysis of end-to-end Internet dynamics'. PhD thesis. University of California, Berkeley, 1997.

[60] Larry Peterson and Timothy Roscoe. 'The Design Principles of PlanetLab'. In: *SIGOPS Oper. Syst. Rev.* 40.1 (Jan. 2006), pp. 11–16. ISSN: 0163-5980. DOI: 10.1145/1113361.1113367. URL: http://doi.acm.org/10.1145/1113361.1113367.

[61] *Ping manpage*. URL: http://man.cx/?page=ping(8).

[62] *Ping (networking utility)*. Mar. 2014. URL: http://en.wikipedia.org/wiki/Ping_(networking_utility).

[63] J. Postel. *Internet Control Message Protocol*. RFC 777. Obsoleted by RFC 792. Internet Engineering Task Force, Apr. 1981. URL: http://www.ietf.org/rfc/rfc777.txt.

[64] J. Postel. *User Datagram Protocol*. RFC 768 (Standard). Internet Engineering Task Force, Aug. 1980. URL: http://www.ietf.org/rfc/rfc768.txt.

[65] John Postel. *Transmission Control Protocol*. RFC 793. Internet Engineering Task Force, Sept. 1981, p. 85. URL: http://www.rfc-editor.org/rfc/rfc793.txt.

[66] Jon Postel, ed. *RFC 791 Internet Protocol - DARPA Inernet Programm, Protocol Specification*. Internet Engineering Task Force. Sept. 1981. URL: http://tools.ietf.org/html/rfc791.

[67] Jon Postel, ed. *RFC 791 Internet Protocol - DARPA Inernet Programm, Protocol Specification*. Internet Engineering Task Force. Sept. 1981. URL: http://tools.ietf.org/html/rfc791.

[68] U. Ranadive and D. Medhi. 'Some observations on the effect of route fluctuation and network link failure on TCP'. In: *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*. 2001, pp. 460–467. DOI: 10.1109/ICCCN.2001.956305.

[69] *RIPE NCC. RIPE NCC Begins to Allocate IPv4 Address Space From the Last /8.* Sept. 2012. URL: http://www.ripe.net/internet-coordination/news/announcements/ripe-ncc-begins-to-allocate-ipv4-address-space-from-the-last-8.

[70] E. Rosen, A. Viswanathan and R. Callon. *Multiprotocol Label Switching Architecture*. RFC 3031 (Proposed Standard). Internet Engineering Task Force, Jan. 2001. URL: http://www.ietf.org/rfc/rfc3031.txt.

[71] Stefan Savage. 'Sting: A TCP-based Network Measurement Tool.' In: *USENIX Symposium on Internet Technologies and Systems*. Vol. 2. 1999, pp. 7–7.

[72] Max Schuchard et al. 'Losing Control of the Internet: Using the Data Plane to Attack the Control Plane'. In: *Proceedings of the 17th ACM Conference on Computer and Communications Security*. CCS '10. Chicago, Illinois, USA: ACM, 2010, pp. 726–728. ISBN: 978-1-4503-0245-6. DOI: 10.1145/1866307.1866411. URL: http://doi.acm.org/10.1145/1866307.1866411.

[73] Neil Spring, Ratul Mahajan and David Wetherall. 'Measuring ISP topologies with Rocketfuel'. In: *ACM SIGCOMM Computer Communication Review* 32.4 (2002), pp. 133–145.

[74] S. Thomson, T. Narten and T. Jinmei. *IPv6 Stateless Address Autoconfiguration*. RFC 4862 (Draft Standard). Internet Engineering Task Force, Sept. 2007. URL: http://www.ietf.org/rfc/rfc4862.txt.

[75] *Usage of IPv6 for websites*. Mar. 2014. URL: http://w3techs.com/technologies/details/ce-ipv6/all/all.

[76] Lindsay Van Eden. *The Truth About ICMP*. 2001.

[77] Darryl Veitch et al. 'Failure control in multipath route tracing'. In: *INFOCOM 2009, IEEE*. IEEE. 2009, pp. 1395–1403.

[78] *V.Jacobson, traceroute*. 1989. URL: ftp://ftp.ee.lbl.gov/.

[79] Feng Wang et al. 'A Measurement Study on the Impact of Routing Events on End-to-end Internet Path Performance'. In: *SIGCOMM Comput. Commun. Rev.* 36.4 (Aug. 2006), pp. 375–386. ISSN: 0146-4833. DOI: 10.1145/1151659.1159956. URL: http://doi.acm.org/10.1145/1151659.1159956.

[80] Jiang Wei-hua, Li Wei-hua and Du Jun. 'The application of ICMP protocol in network scanning'. In: *Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on*. IEEE. 2003, pp. 904–906.

[81] Annika Wennström, Stefan Alfredsson and Anna Brunstrom. 'TCP over wireless networks'. In: (2004).

[82] Peng Wu et al. 'Transition from IPv4 to IPv6: A State-of-the-Art Survey'. In: *Communications Surveys Tutorials, IEEE* 15.3 (Third 2013), pp. 1407–1424. ISSN: 1553-877X. DOI: 10.1109/SURV.2012.110112.00200.

[83] Sebastian Zander and Grenville Armitage. 'Minimally-intrusive frequent round trip time measurements using Synthetic Packet-Pairs'. In: *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*. Oct. 2013, pp. 264–267. DOI: 10.1109/LCN.2013.6761245.